



Ersilia



H3D Foundation and Ersilia present

Bringing data science and AI/ML tools to
infectious disease research

Event Sponsors



CS&S

Code for
Science
& Society





Session 2: virtual screening cascade

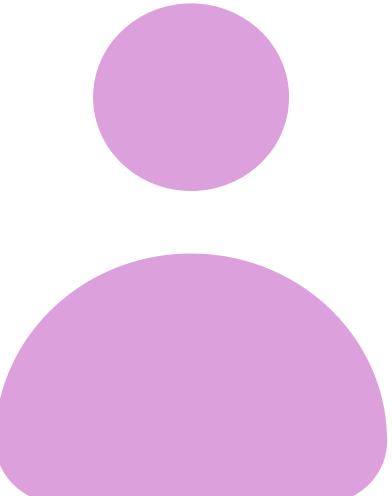
Breakout session

Gemma Turon, @TuronGemma, gemma@ersilia.io

Miquel Duran-Frigola, @mduranfrigola, miquel@ersilia.io

Ersilia Open Source Initiative, @ersiliaio, <https://ersilia.io>

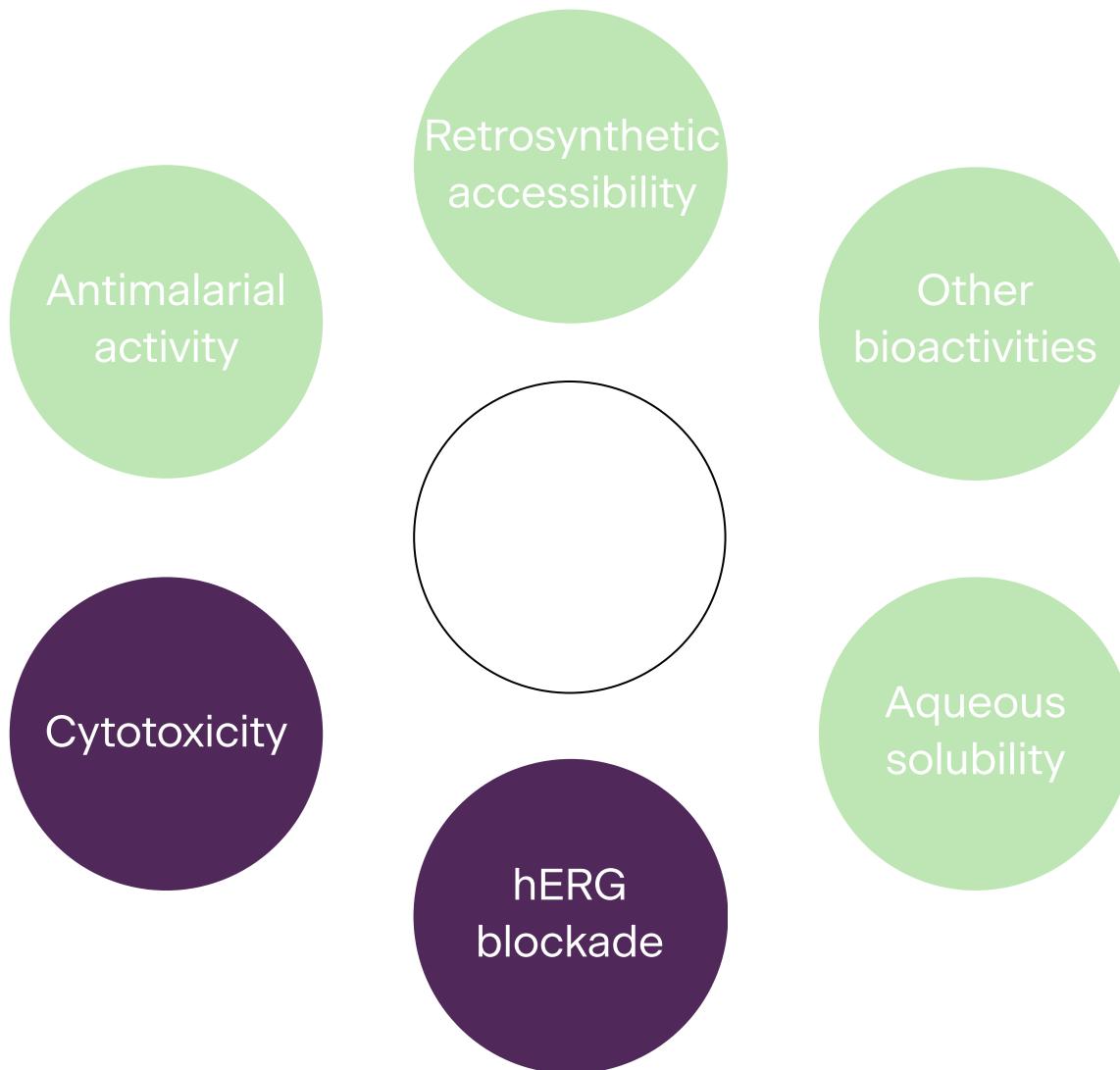
28th September 2022



I have access to a library of compounds for only their structure (SMILES) is available. I want to identify a few hits for an anti-infective drug, but only have the capacity to test 10 molecules in the first round. What can I do?

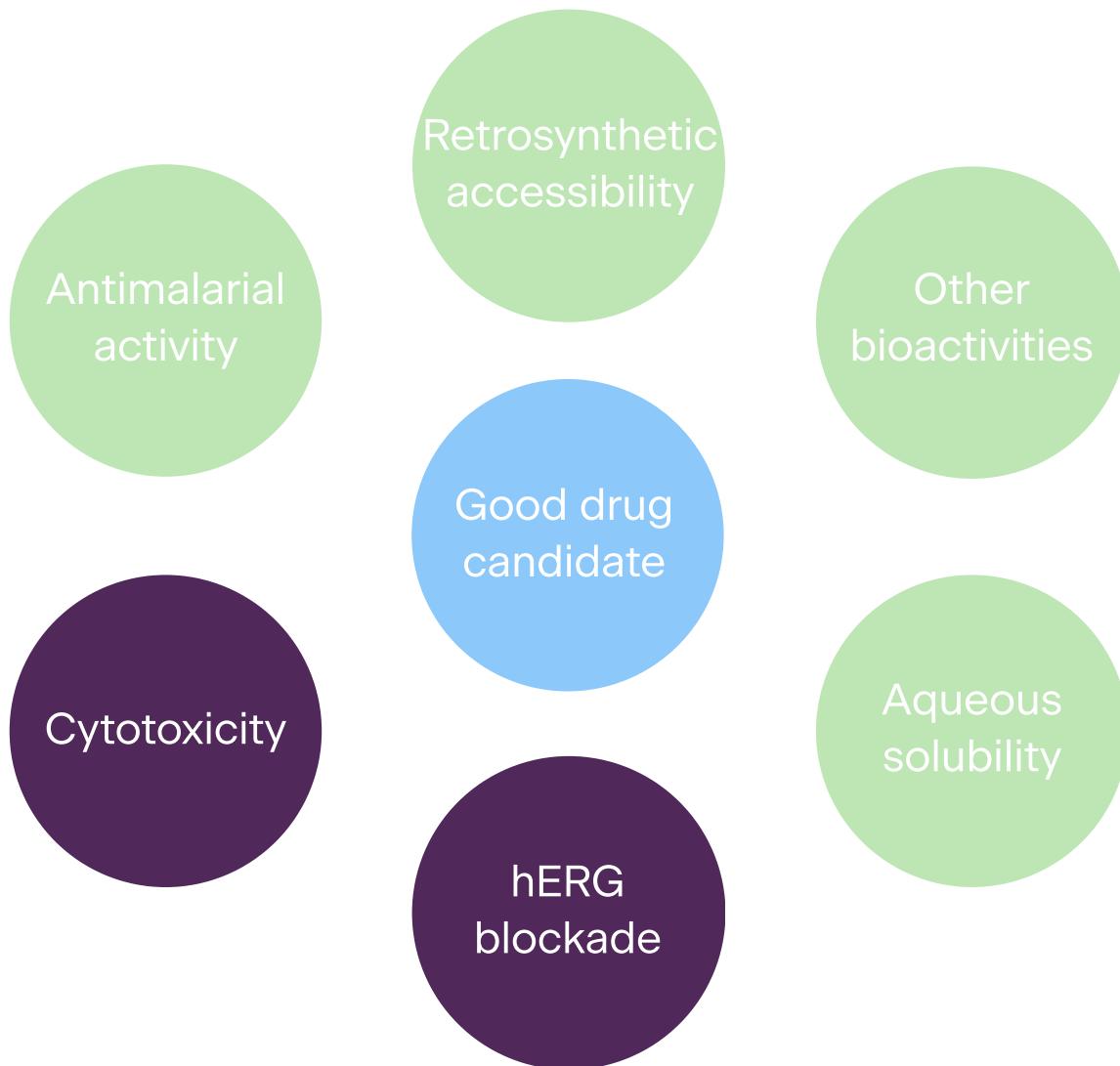


Virtual Screening Cascade



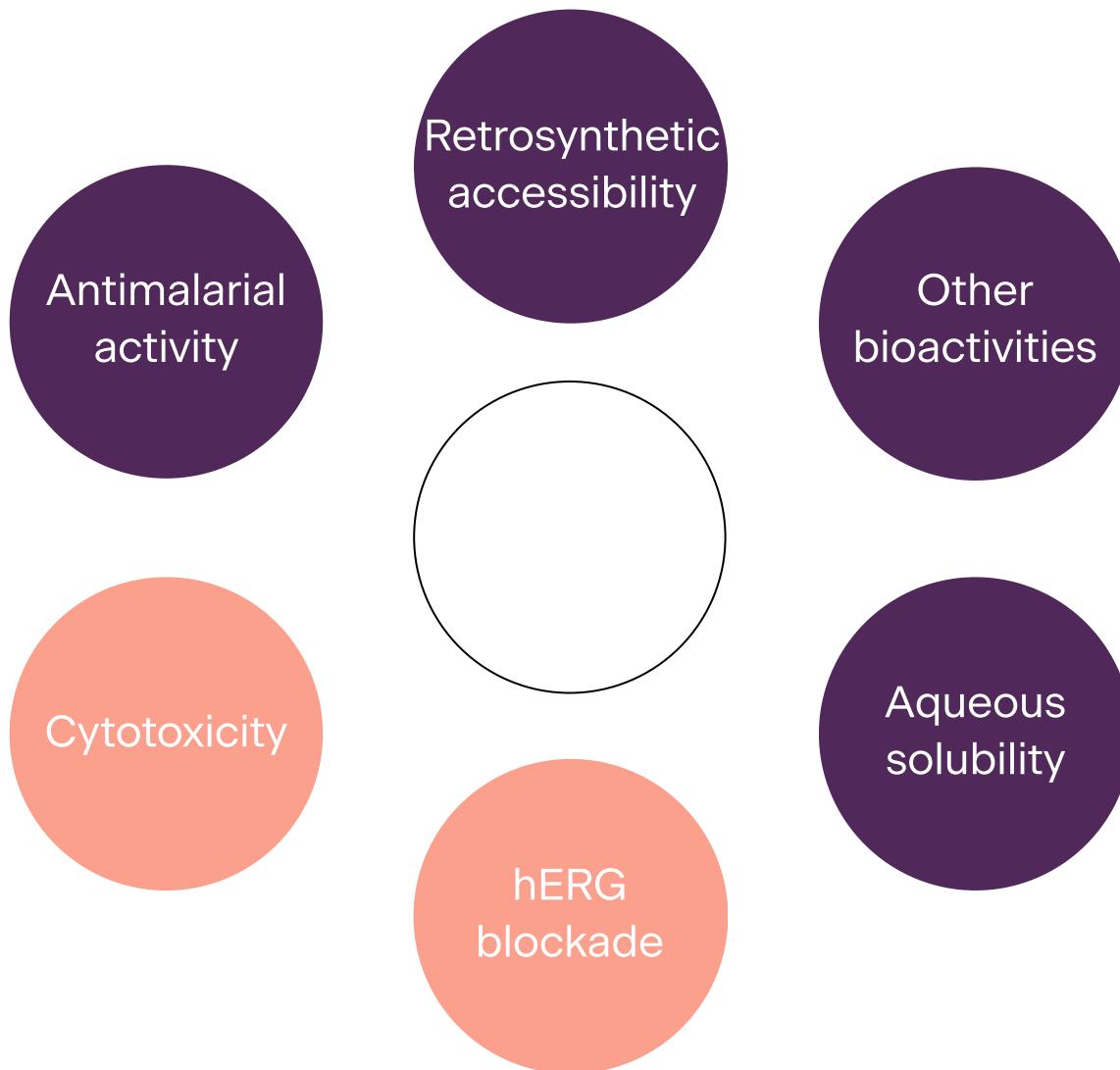


Virtual Screening Cascade



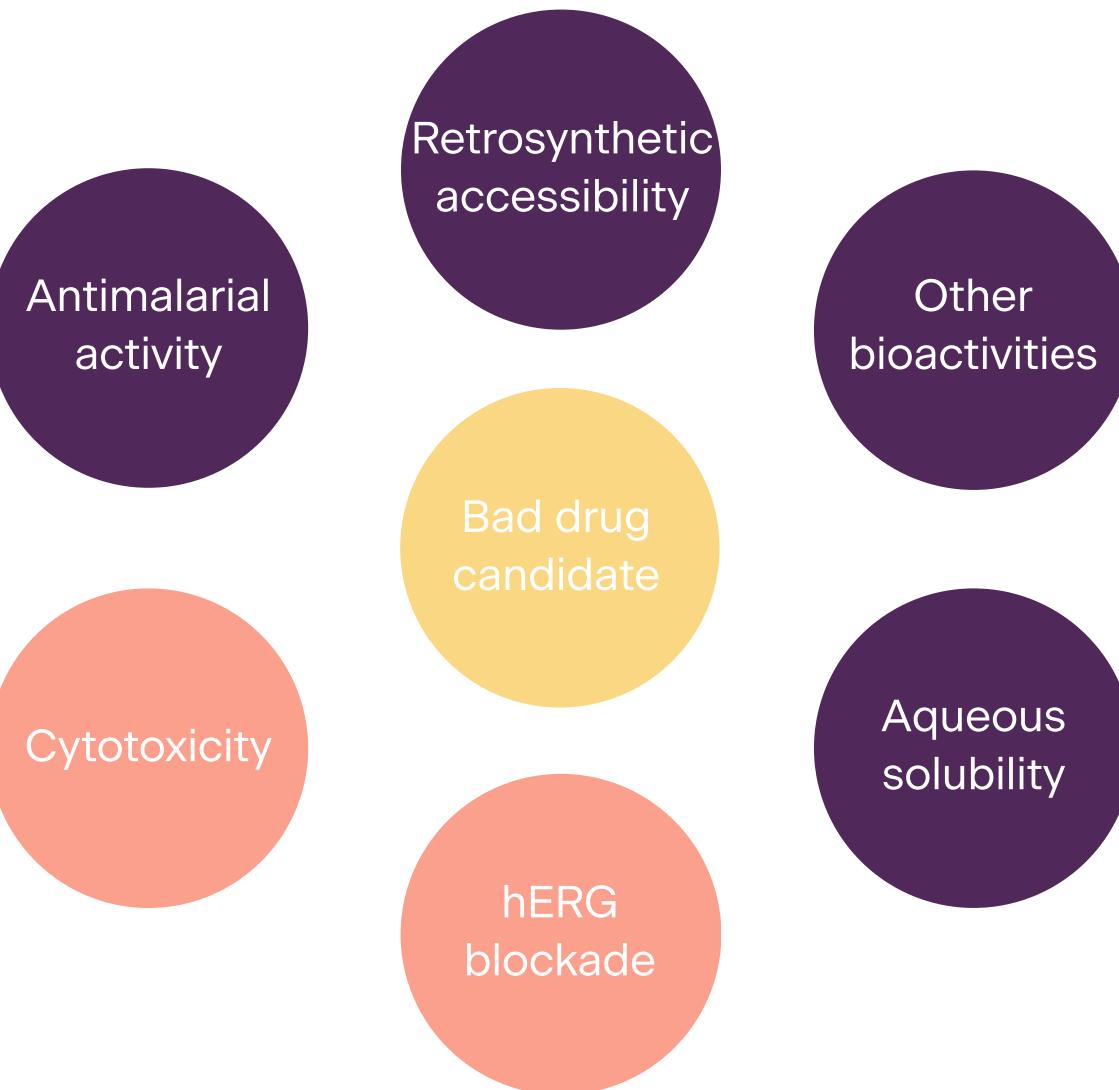


Virtual Screening Cascade





Virtual Screening Cascade





Ersilia Model Hub

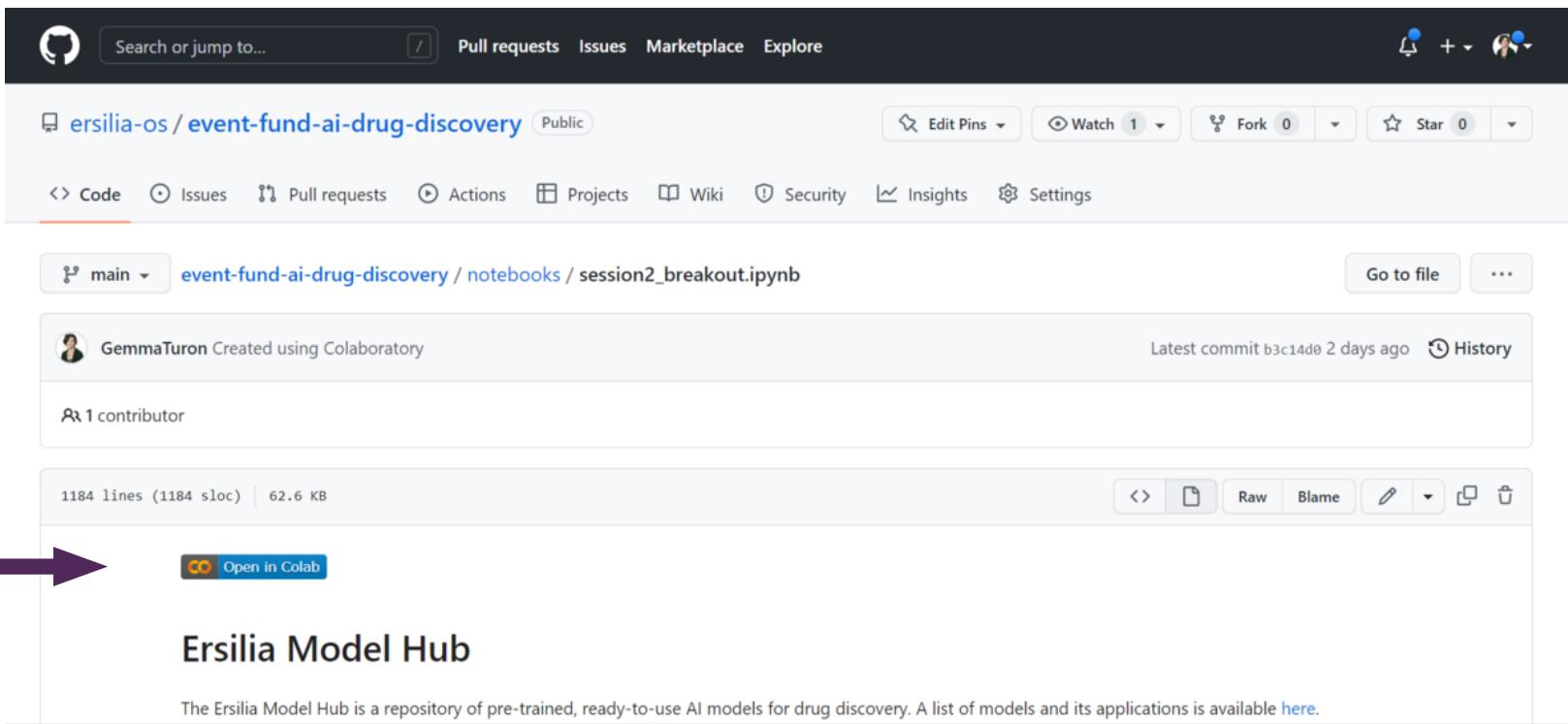
Repository of pre-trained, ready to use AI/ML models for drug discovery

- Available models can be browsed at <https://ersilia.io/model-hub>
- The code is available at <https://github.com/ersilia-os/ersilia> (GPLv3 License)
- Accessibility:
 - Command Line Interface
 - Google Colab implementation



Getting Started

<https://github.com/ersilia-os/event-fund-ai-drug-discovery>



Search or jump to... Pull requests Issues Marketplace Explore

ersilia-os / [event-fund-ai-drug-discovery](#) Public Edit Pins Watch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main event-fund-ai-drug-discovery / notebooks / session2_breakout.ipynb Go to file ...

GemmaTuron Created using Colaboratory Latest commit b3c14d0 2 days ago History

1 contributor

1184 lines (1184 sloc) | 62.6 KB Raw Blame

Open in Colab

Ersilia Model Hub

The Ersilia Model Hub is a repository of pre-trained, ready-to-use AI models for drug discovery. A list of models and its applications is available [here](#).



Ersilia Model Hub in Colab



Click on the play button to install Ersilia in this Colab notebook.

Show code



```
1 #@title The Ersilia Model Hub
2 #@markdown Click on the play button to install Ersilia in this Colab
notebook.
3
4 %%capture
5 %env MINICONDA_INSTALLER_SCRIPT=Miniconda3-py37_4.12.0-Linux-x86_64.sh
6 %env MINICONDA_PREFIX=/usr/local
7 %env PYTHONPATH={PYTHONPATH}:/usr/local/lib/python3.7/site-packages
8 %env CONDA_PREFIX=/usr/local
9 %env CONDA_PREFIX_1=/usr/local
10 %env CONDA_DIR=/usr/local
11 %env CONDA_DEFAULT_ENV=base
12 !wget https://repo.anaconda.com/miniconda/$MINICONDA_INSTALLER_SCRIPT
13 !chmod +x $MINICONDA_INSTALLER_SCRIPT
14 !./$MINICONDA_INSTALLER_SCRIPT -b -f -p $MINICONDA_PREFIX
15 !python -m pip install git+https://github.com/ersilia-os/ersilia.git
16 !python -m pip install requests --upgrade
17 import sys
18 _ = (sys.path.append("/usr/local/lib/python3.7/site-packages"))
```



Ersilia Model Hub

Each model is identified by a code (eosxxxx) and a slug (1-2 word reference).

There are five basic commands on Ersilia:

1. Fetch the model from its online storage
2. Serve the model on your system
3. Run the desired API (predict)
4. Close the model (stop serving)
5. Delete the model from the system



Breakout Session Exercise

We will use the 400 compounds from the MMV Malaria Box to run a series of predictions using models available in the Ersilia Model Hub and select the molecules with higher interest for experimental testing.

Steps:

1. Install the Ersilia Model Hub in Colab (together)
2. Run a model for antimalarial activity prediction and analyse the results (together)
3. In small groups, run and analyse the outcomes of the different suggested models



Antimalarial Activity

Tags

- Tox21
- Toxicity
- MoleculeNet
- Grover
- Graph Transformer

Output

- Antibiotic activity
- Toxicity
- Synthetic accessibility
- Antiviral activity
- Target

Mode

Antimalarial activity prediction

Prediction of the antimalarial potential of small molecules. Originally trained on a proprietary dataset of >7M compounds and retrained using ChEMBL21 (2M compounds)

eos2gth

maip-malaria-surrogate

Date of incorporation
11/18/2021

Publication
<https://jcheminf.biomedcentral.com/articles/10.1186/s13321-021-00487-2>

Original code
<https://www.ebi.ac.uk/chembl/maip/>

Try it with the Ersilia CLI!

```
$ ersilia serve maip-malaria-surrogate
$ ersilia api -i 'CCCOCCC'
$ ersilia close
```



Mounting Google Drive

We will first mount google drive on Colab to store the model predictions and import some basic Python Packages

```
● ● ●  
1 #mount your own GDrive in Colab  
2 from google.colab import drive  
3 drive.mount('/content/drive')  
4  
5 import matplotlib.pyplot as plt  
6 import pandas as pd
```



MMV Malaria Box

The dataset is prepared and stored as a csv file in the /data folder of the h3d_ersilia_ai_workshop in your personal Drive



```
1 #we can open it as a pandas dataframe
2 smiles = "drive/MyDrive/h3d_ersilia_ai_workshop/data/session2/mmv_malariabox.csv"
3 df=pd.read_csv(smiles)
4 df.head()
```

	CAN_SMILES	edit
0	COc1ccccc1CNC(=O)CCn1c(=O)[nH]c2ccsc2c1=O	
1	CN(C)c1ccc(C(O)(c2ccc(N(C)C)cc2)c2ccc(N(C)C)cc...	
2	Cc1ccc(-c2cc3c(SCC(=O)Nc4cc(C(F)(F)F)ccc4Cl)nc...	
3	CCOC(=O)C1=C(c2cccc2)N=c2sc(=Cc3cc(C)n(-c4ccc...	
4	CCOC(=O)c1cnc2c(C)cc(C)cc2c1Nc1ccc(OC)c(OC)c1	



Antimalarial Activity: eos2gth

We use the ! notation to access the CLI commands for Ersilia from the notebook

Data is prepared in the folder /data already in your google drive, and predictions will also be stored there.

```
● ● ●  
1 #Step One: Retrieve the model from the internet  
2 !ersilia fetch eos2gth
```

```
! Fetching model eos2gth: maip-malaria-surrogate  
👍 Model eos2gth fetched successfully!
```



Antimalarial Activity: eos2gth



```
1 #we must import the ErsiliaModel Function for Python
2 from ersilia import ErsiliaModel
3
4 #Step 2: load the model in the notebook
5 model = ErsiliaModel("eos2gth")
6 #Step 3: bring the model alive
7 model.serve()
8 #Step 4: run predictions for the input smiles
9 output = model.predict(input=smiles, output="pandas")
10 #Step 5: close the model
11 model.close()
```

We have obtained the predictions as "output" in pandas format, which we can directly save to Drive as .csv



```
1 output.to_csv("drive/MyDrive/DataScience_Workshop/data/day2/eos2gth.csv",
 index=False)
```



Analysing predictions

To analyse the predictions, we can read them from the Drive



```
1 df = pd.read_csv("drive/MyDrive/DataScience_Workshop/data/day2/eos2gth.csv")
2 df.head()
```

	key	input	score
0	ALGPHOUNWIZIOQ-UHFFFAOYSA-N	COc1ccccc1CNC(=O)CCn1c(=O)[nH]c2ccsc2c1=O	6.886159
1	QFVDKARCPMTZCS-UHFFFAOYSA-N	CN(C)c1ccc(C(O)(c2ccc(N(C)C)cc2)c2ccc(N(C)C)cc...	15.483176
2	HKNNPGWJKJDXCN-UHFFFAOYSA-N	Cc1ccc(-c2cc3c(SCC(=O)Nc4cc(C(F)(F)F)ccc4Cl)nc...	27.288107
3	QQNUVMRXVVLAAU-UHFFFAOYSA-N	CCOC(=O)C1=C(c2cccc2)N=c2sc(=Cc3cc(C)n(-c4ccc...	12.532308
4	MSGARPVCYTSLR-UHFFFAOYSA-N	CCOC(=O)c1cnc2c(C)cc(C)cc2c1Nc1ccc(OC)c(OC)c1	37.554440

InChiKey

SMILES

Prediction



Analysing predictions

We can sort the molecules from higher to lower score



```
1 output.sort_values("score", ascending=False).head()
```

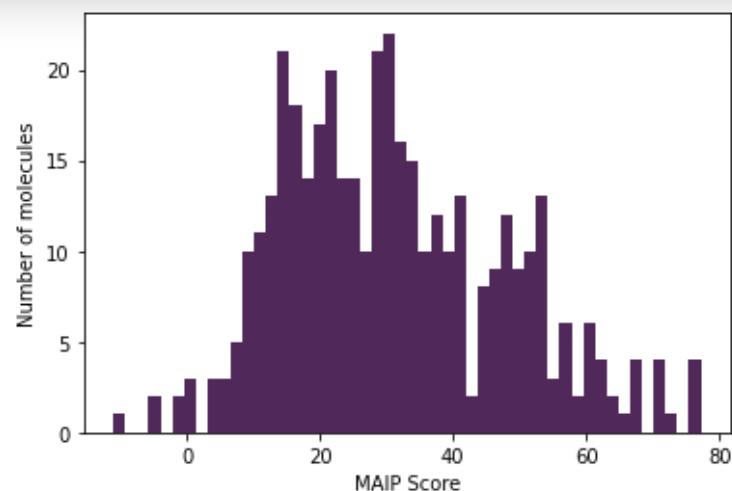
		key	input	score
136		HPFVQAYLQOSFOL-UHFFFAOYSA-N	COc1cccc(Nc2nc(NCCO)c3cccc3n2)c1	77.199020
399		ZGMMVVYGDFQTBB-UHFFFAOYSA-N	OCCNc1nc(Nc2ccc(Cl)c(Cl)c2)nc2cccc12	76.798370
69		RHZLKBRFIAZMTN-UHFFFAOYSA-N	Cc1ccc(Nc2nc(NCCO)c3cccc3n2)cc1C	76.052498
339		NPWXHTXMBIOHKI-UHFFFAOYSA-N	Cn1c(=O)n(C)c2cc(CNCCNc3ccnc4cc(Cl)ccc34)ccc21	76.012589
55		YZZGEZJZYYZGG-UHFFFAOYSA-N	CCN(CC)CCNc1ncnc2c1[nH]c1ccc(Cl)cc12	71.951157



Analysing predictions

Or plot the distribution of all scores in a histogram

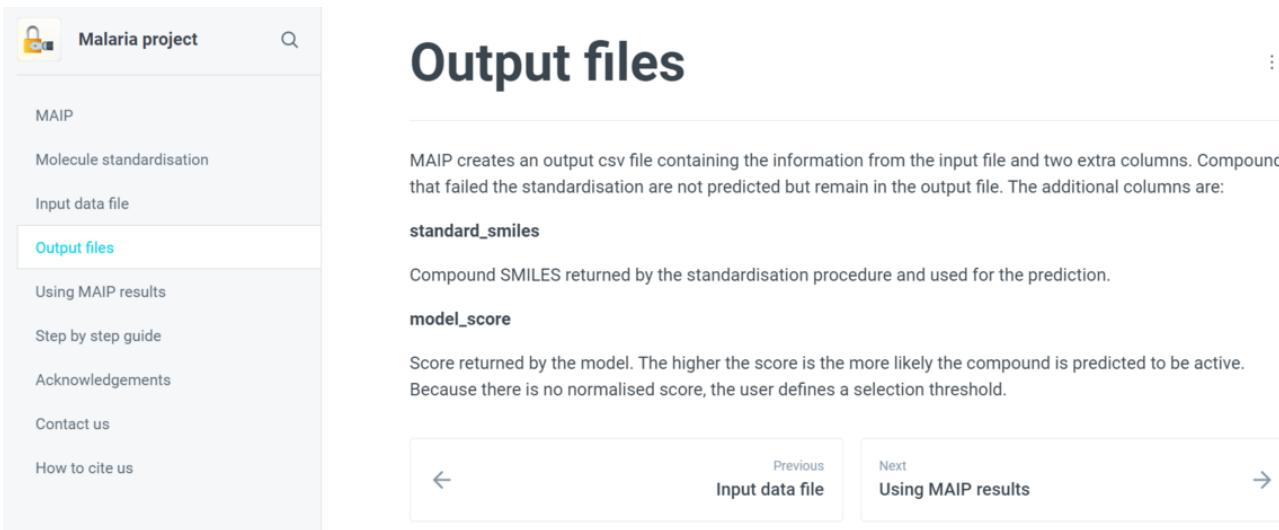
```
● ● ●  
1 plt.hist(output["score"], bins=50, color="#50285a")  
2 plt.xlabel("MAIP Score")  
3 plt.ylabel("Number of molecules")  
4 plt.show()
```





Analysing predictions

- Is this model a regression or a classification?
- How can we interpret the model "score" as antimalarial potential?



The screenshot shows a sidebar menu for a 'Malaria project' with the following items:

- MAIP
- Molecule standardisation
- Input data file
- Output files** (highlighted)
- Using MAIP results
- Step by step guide
- Acknowledgements
- Contact us
- How to cite us

The main content area is titled 'Output files' and contains the following text:

MAIP creates an output csv file containing the information from the input file and two extra columns. Compounds that failed the standardisation are not predicted but remain in the output file. The additional columns are:

standard_smiles
Compound SMILES returned by the standardisation procedure and used for the prediction.

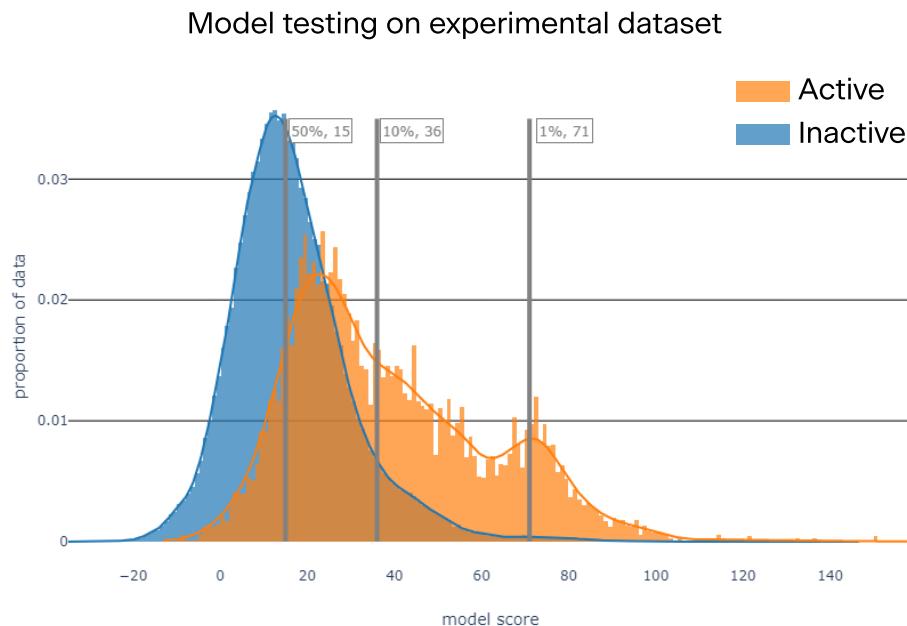
model_score
Score returned by the model. The higher the score is the more likely the compound is predicted to be active. Because there is no normalised score, the user defines a selection threshold.

Navigation buttons at the bottom include arrows for 'Previous' (Input data file) and 'Next' (Using MAIP results).

Plotting the results

To better understand the model outputs, we need to go back to the original model publication where the training data is explained

Given this test set,
what could we
choose as a good
activity threshold
for our dataset?





Next steps

In groups, run additional predictions for the MMV Dataset and select the molecules that you would move on to experimental testing.

We provide a list of suggested models to be used, you do not need to run predictions for all:

- eos46ev (anti tuberculosis)
- eos4e41 (antibiotic activity)
- eos2ta5 (hERG blockade)
- eos2r5a (retrosynthetic availability)
- eos6oli (aqueous solubility)
- eos9yui (natural product likeness)



Next steps

Some pointers:

- Run predictions for relevant models, storing the results in Google Drive
- Analyse the results on the Google Colab Notebook or directly using Excel / Google Sheets
- Discuss the results and rank/select best molecules
- Prepare a short (10 min presentation) of your findings

There is no right or wrong answer, it's just a practise exercise

<https://ersilia.gitbook.io/event-fund/>