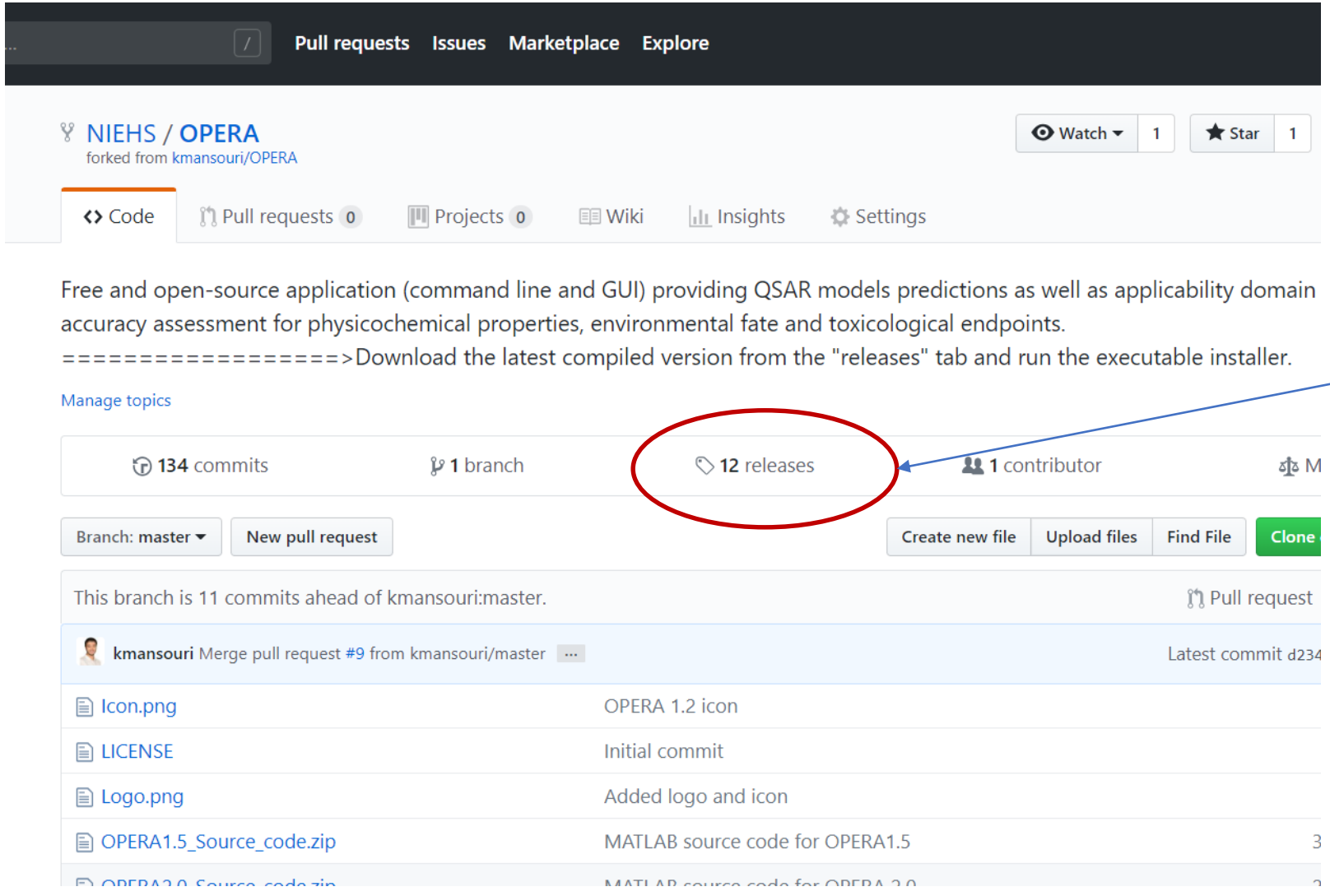
Windows version installation and quick run guide

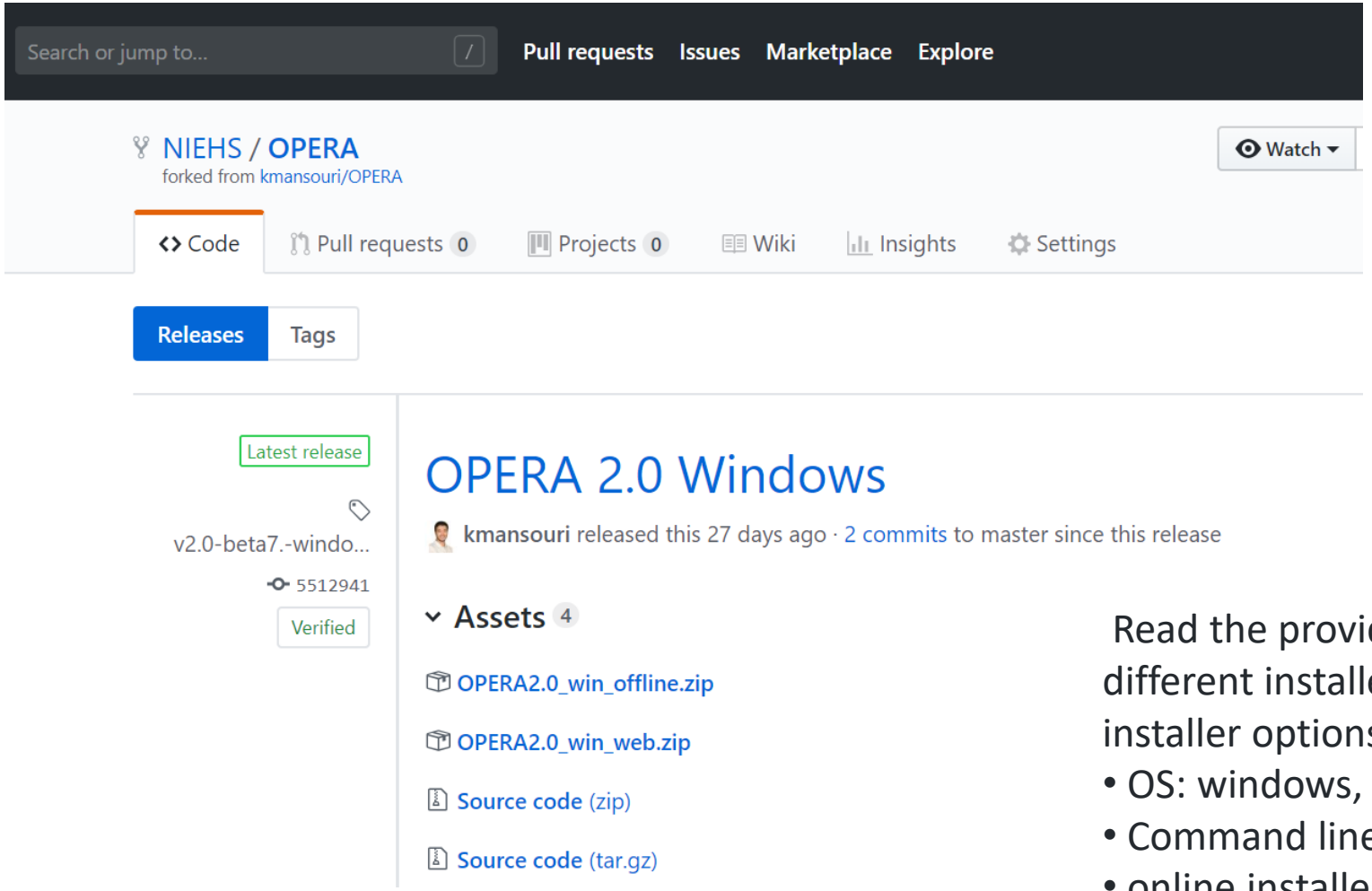
Command line & GUI

OPERA Github repo:



Go to the releases to download the latest version

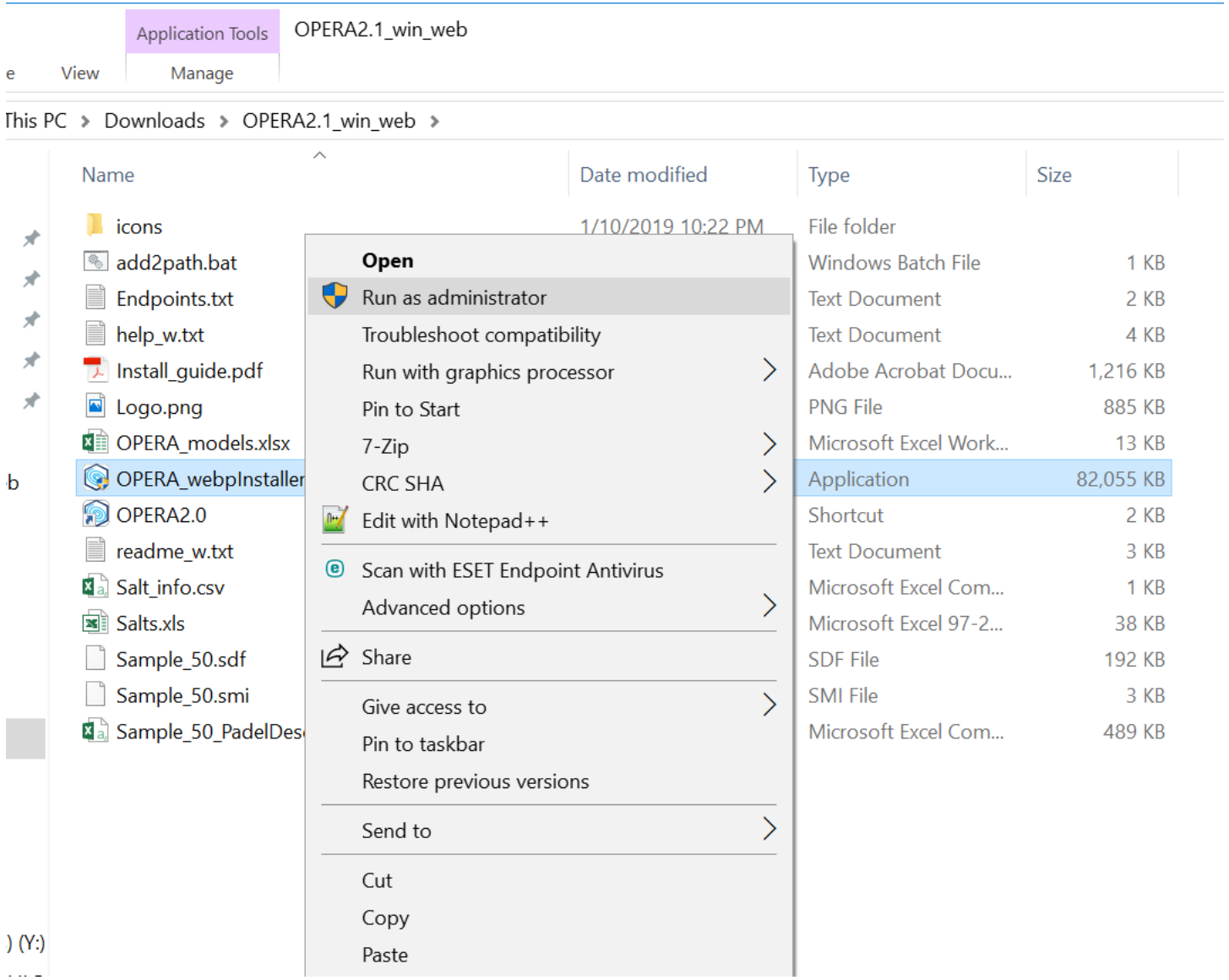
Download the installer from “Releases”



Read the provided information about the different installers available and select one of the installer options:

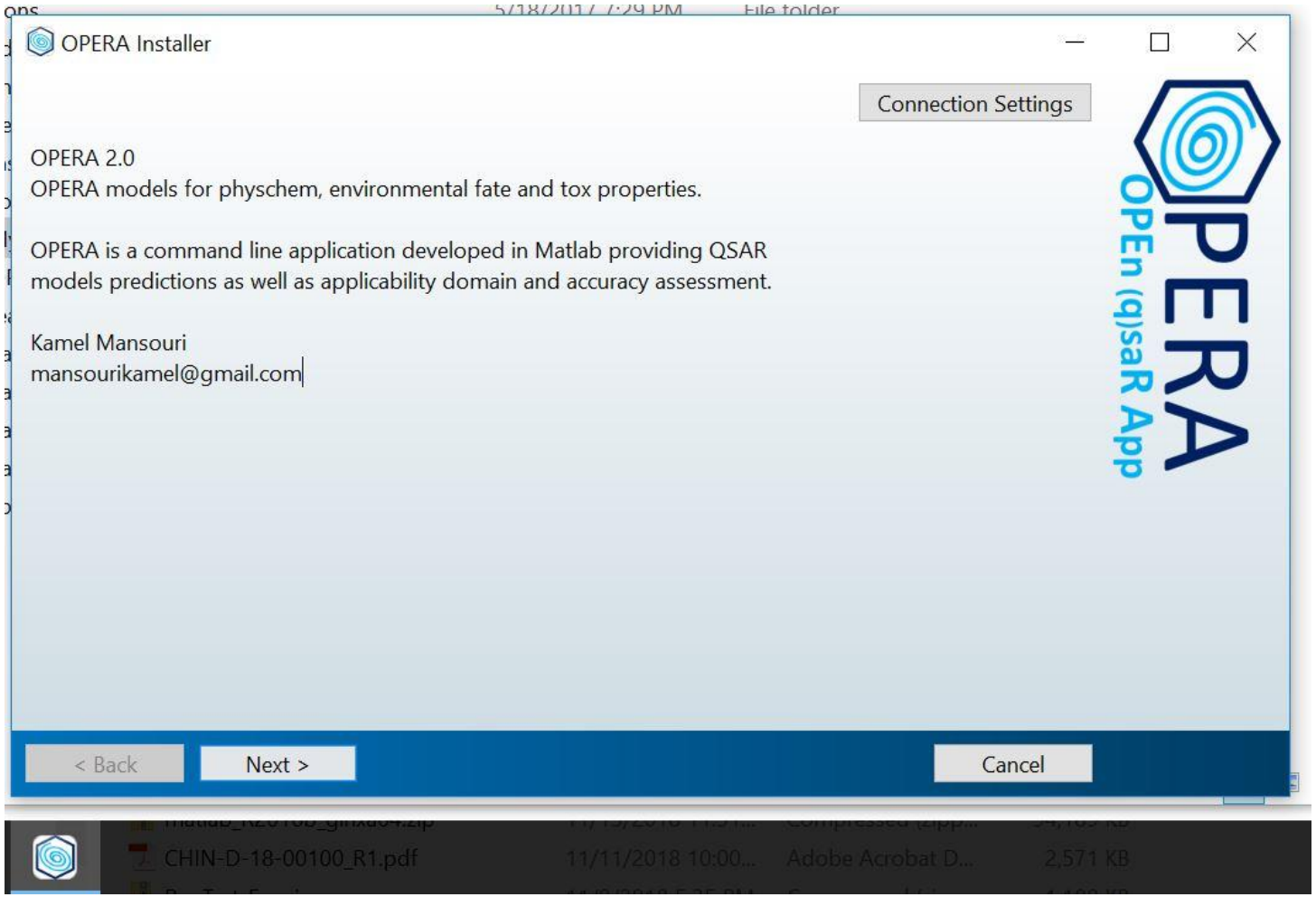
- OS: windows, Linux
- Command line, GUI
- online installer, offline installer

Unzip and run OPERA_Installer as administrator

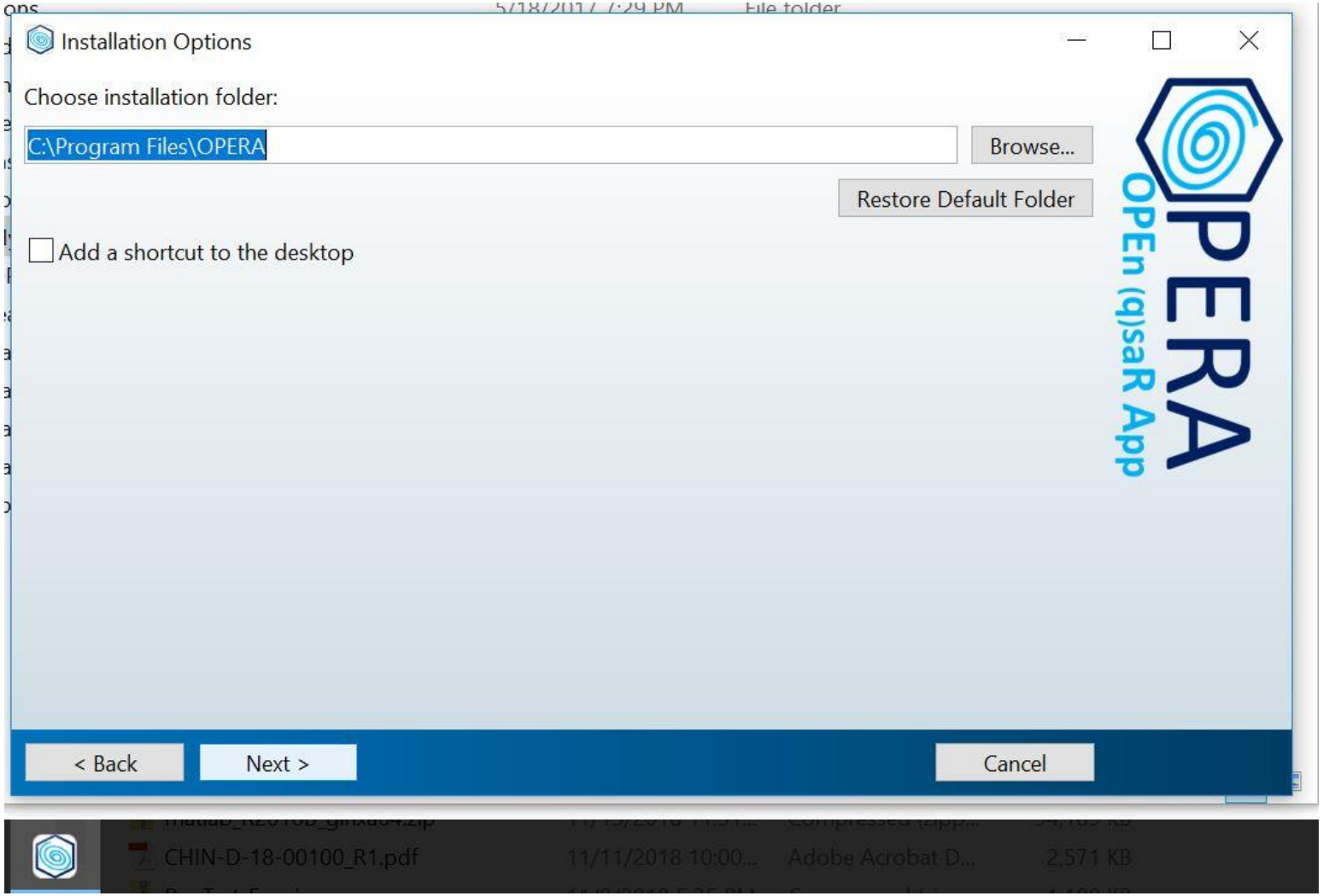


The installer will unpack and initialize quickly

First intro to OPERA, click next when ready



Recommended install options.

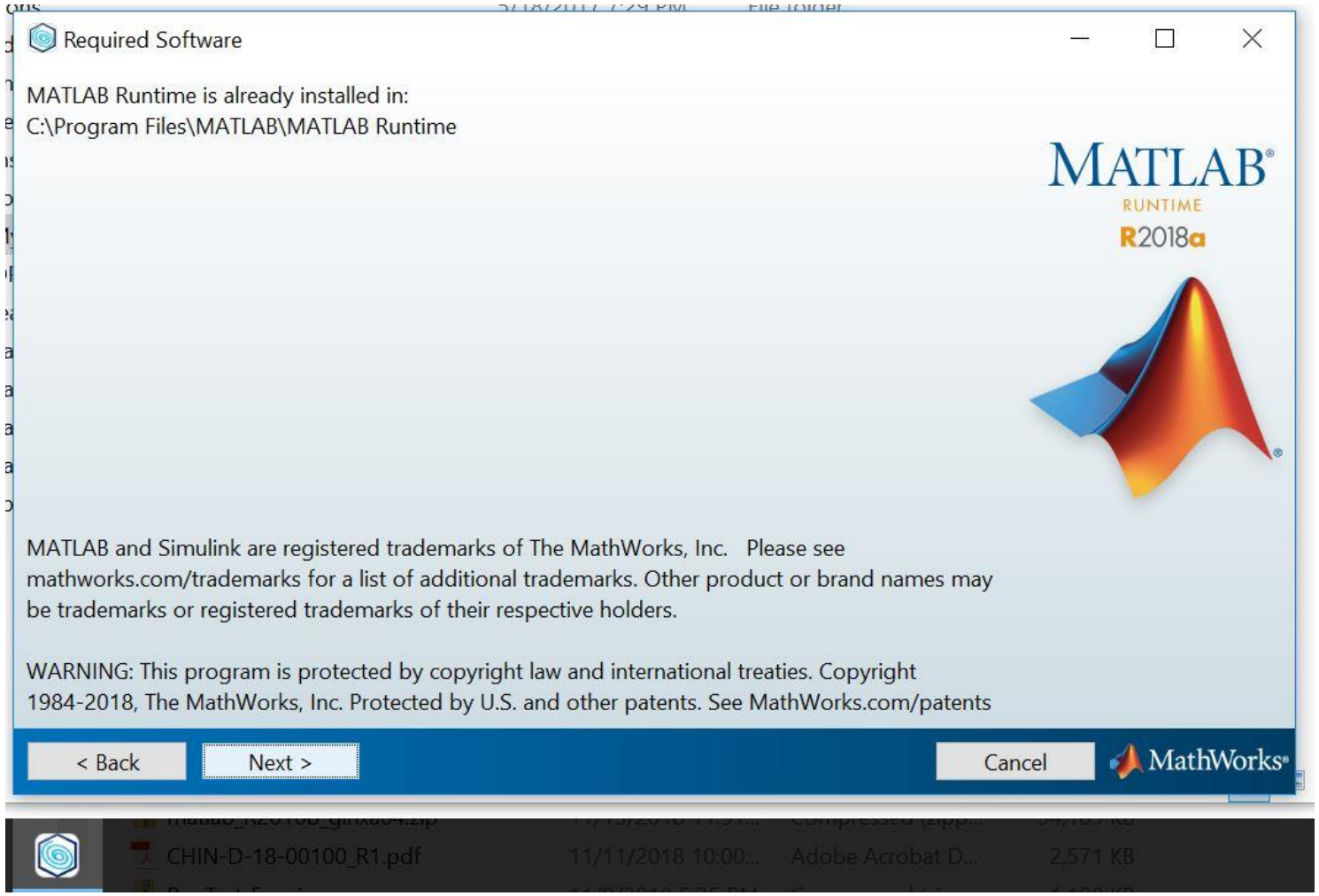


It is important to

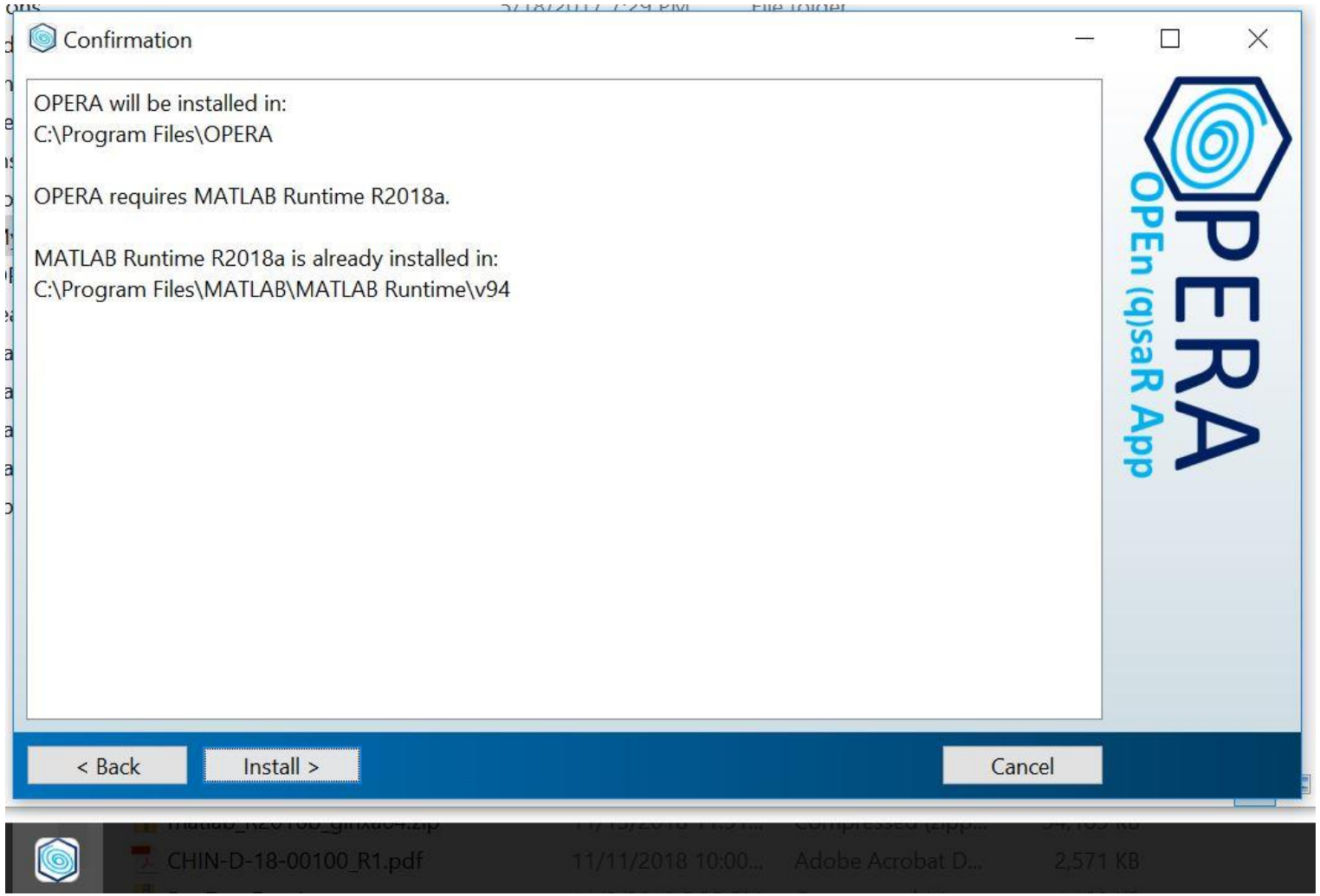
- For GUI, select add shortcut
- For command line, the shortcut is provided in the unzipped file. Just copy paste it to the desktop.

- Note that if the default installation folder is modified:
- The GUI will adapt automatically
- The command line shortcut location needs to be updated and further instructions will be provided during the first run.

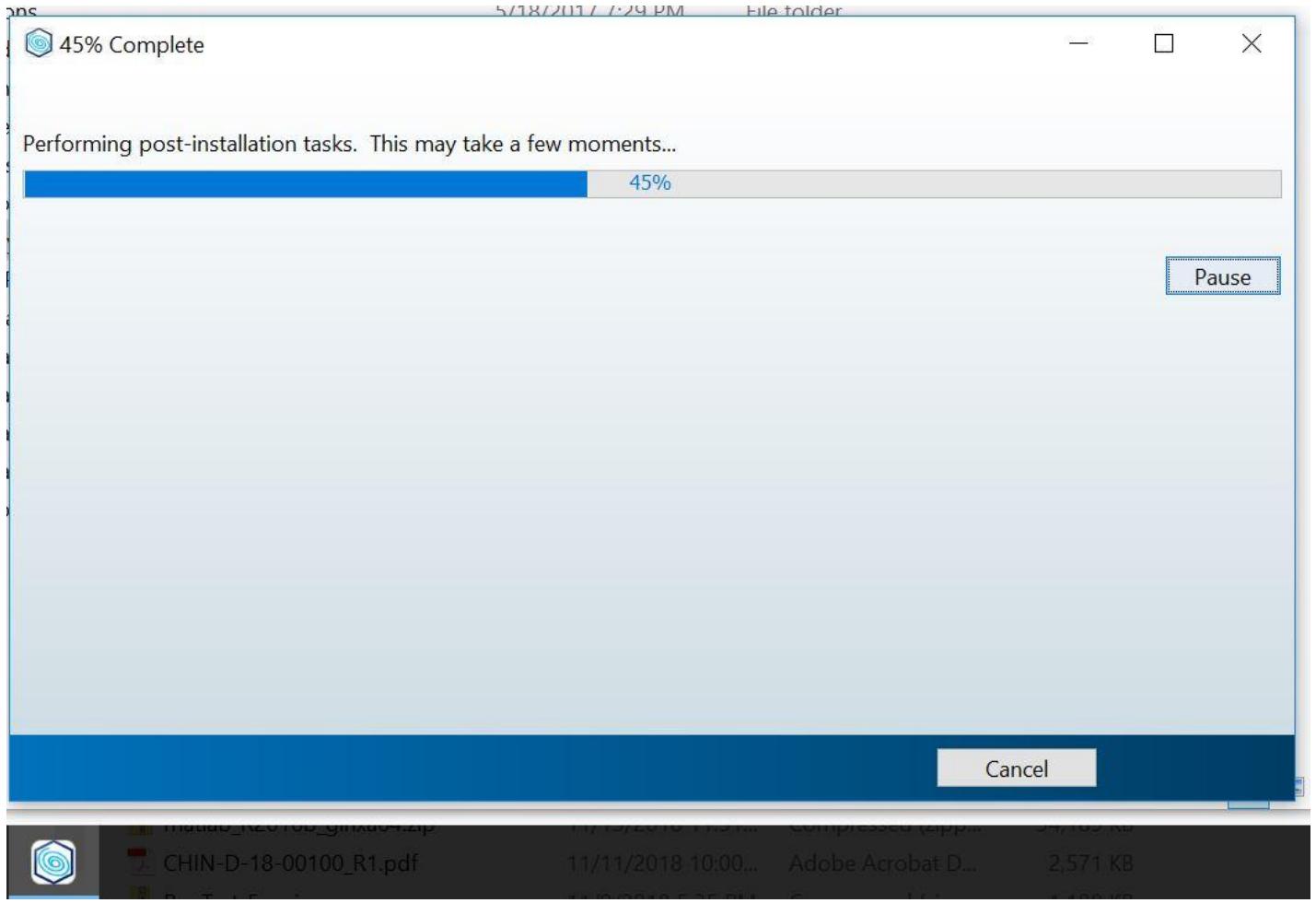
Just click next



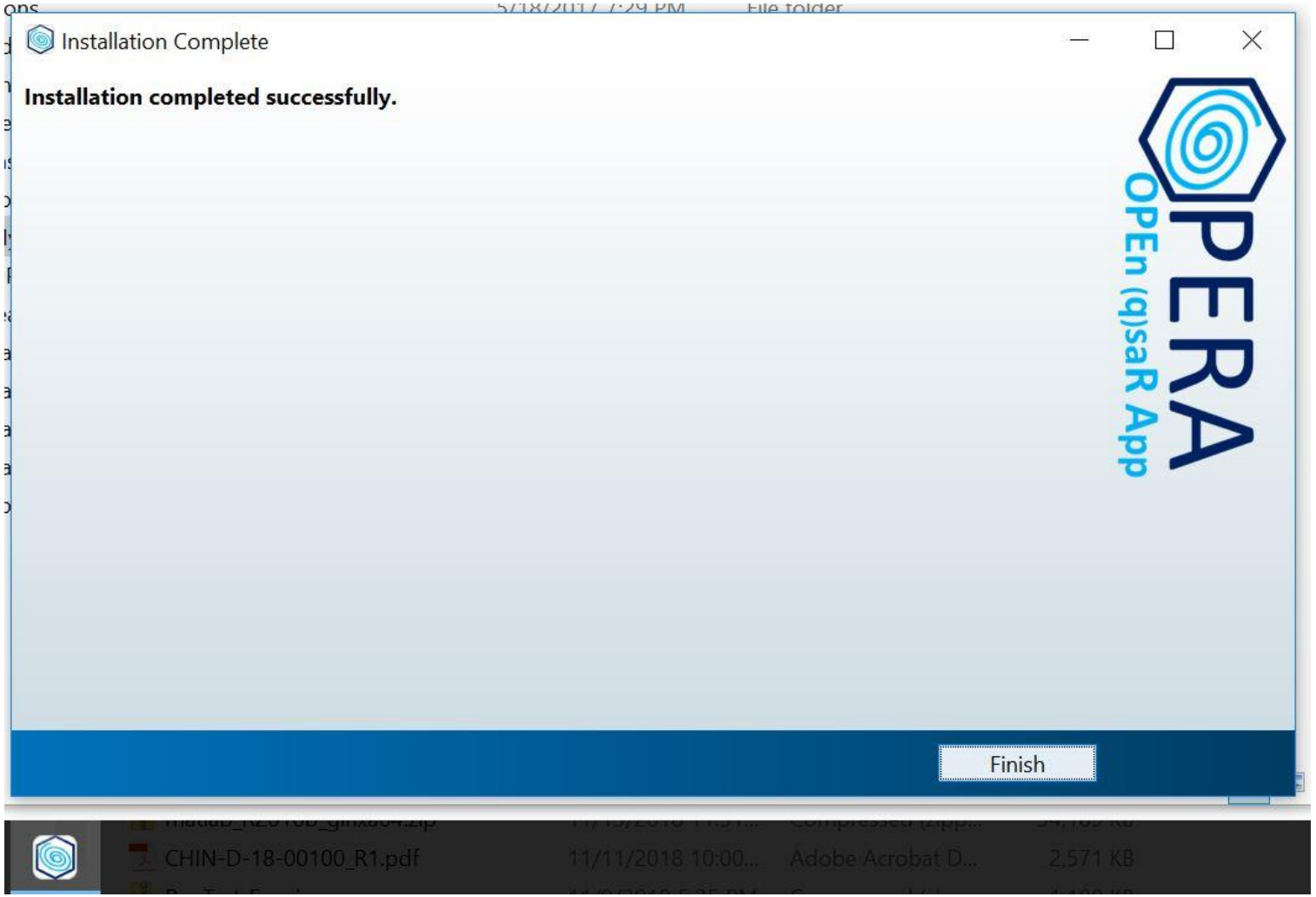
Confirm and install.



Wait while downloading and/or installing the runtime. (next time will go faster)

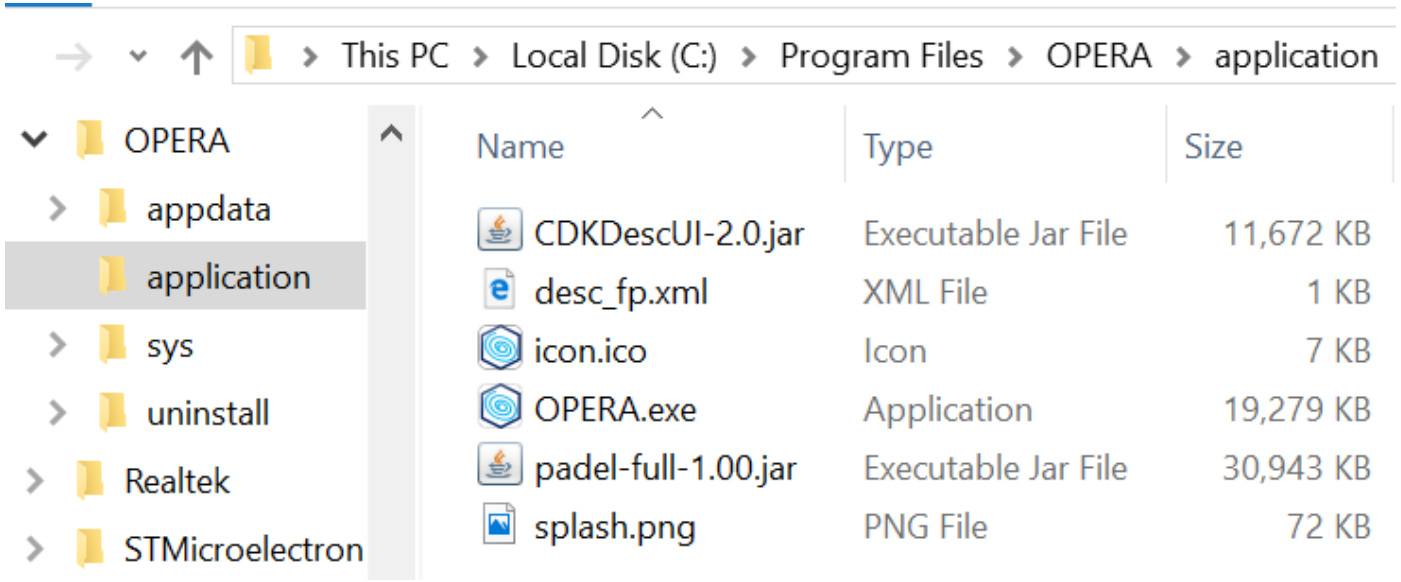


That’s it! IT’s OPERAtional.

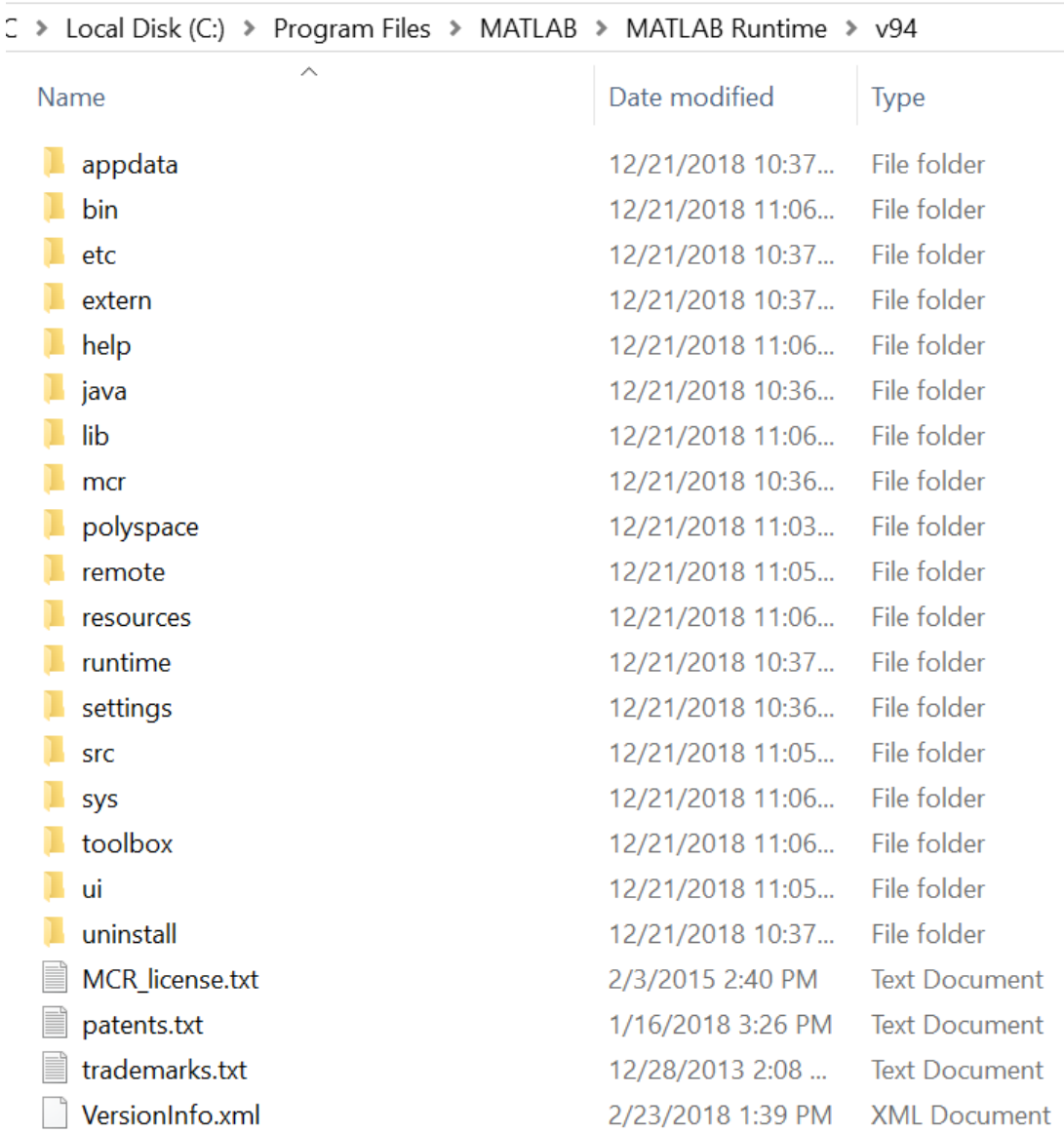


Installed files

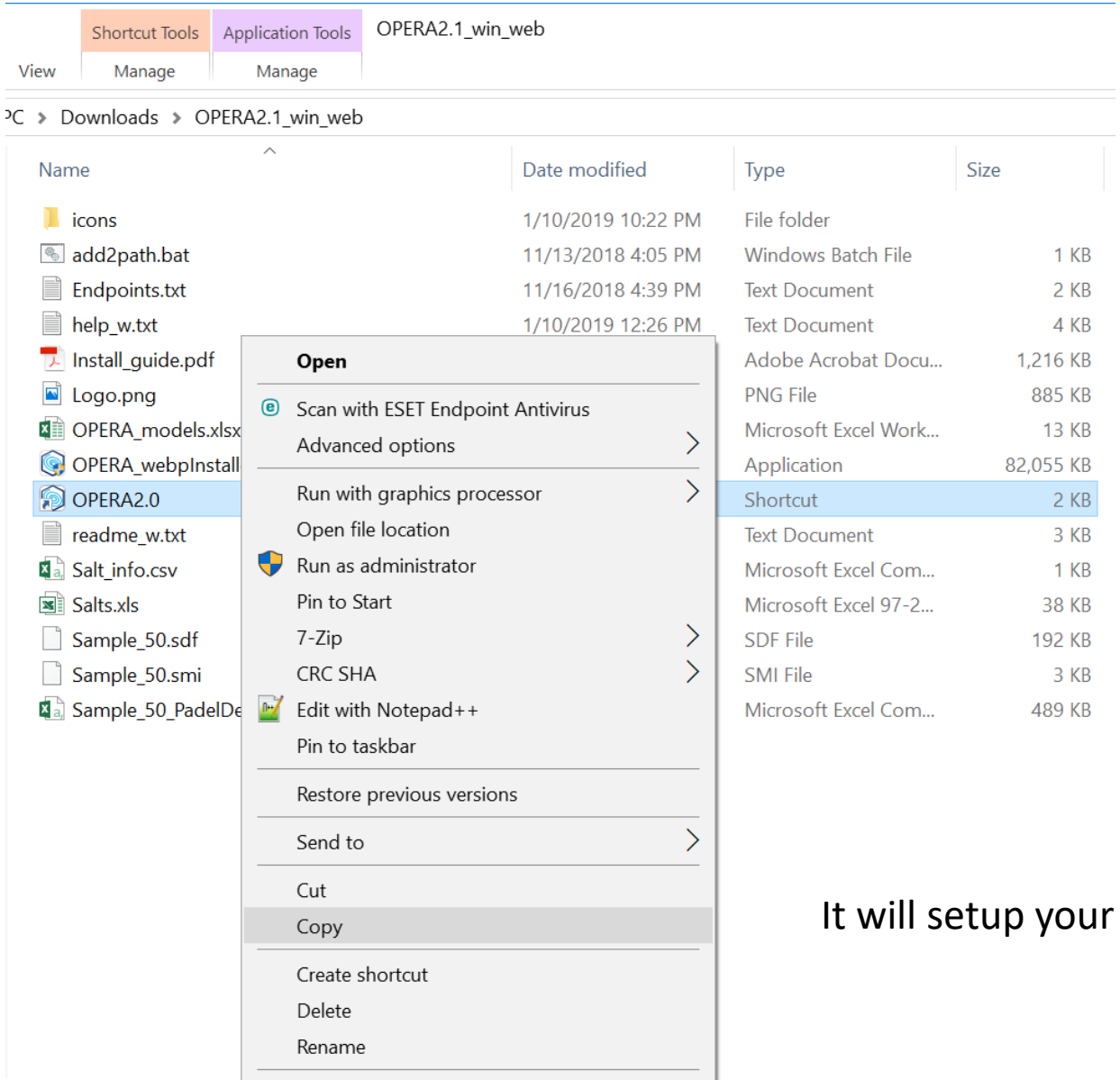
OPERA files

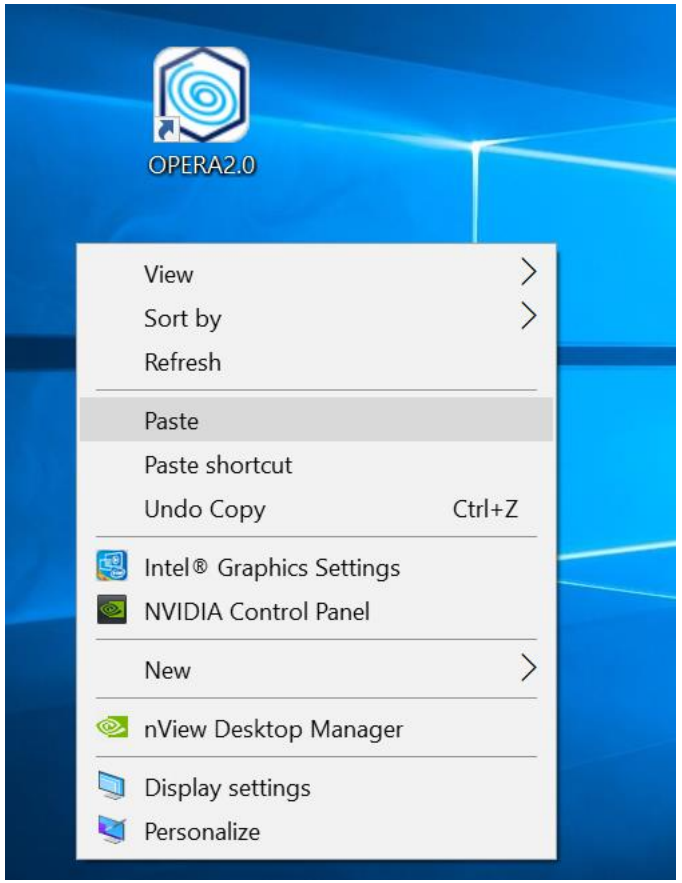


Runtime files



For command line only:
copy the provided shortcut to your desktop.





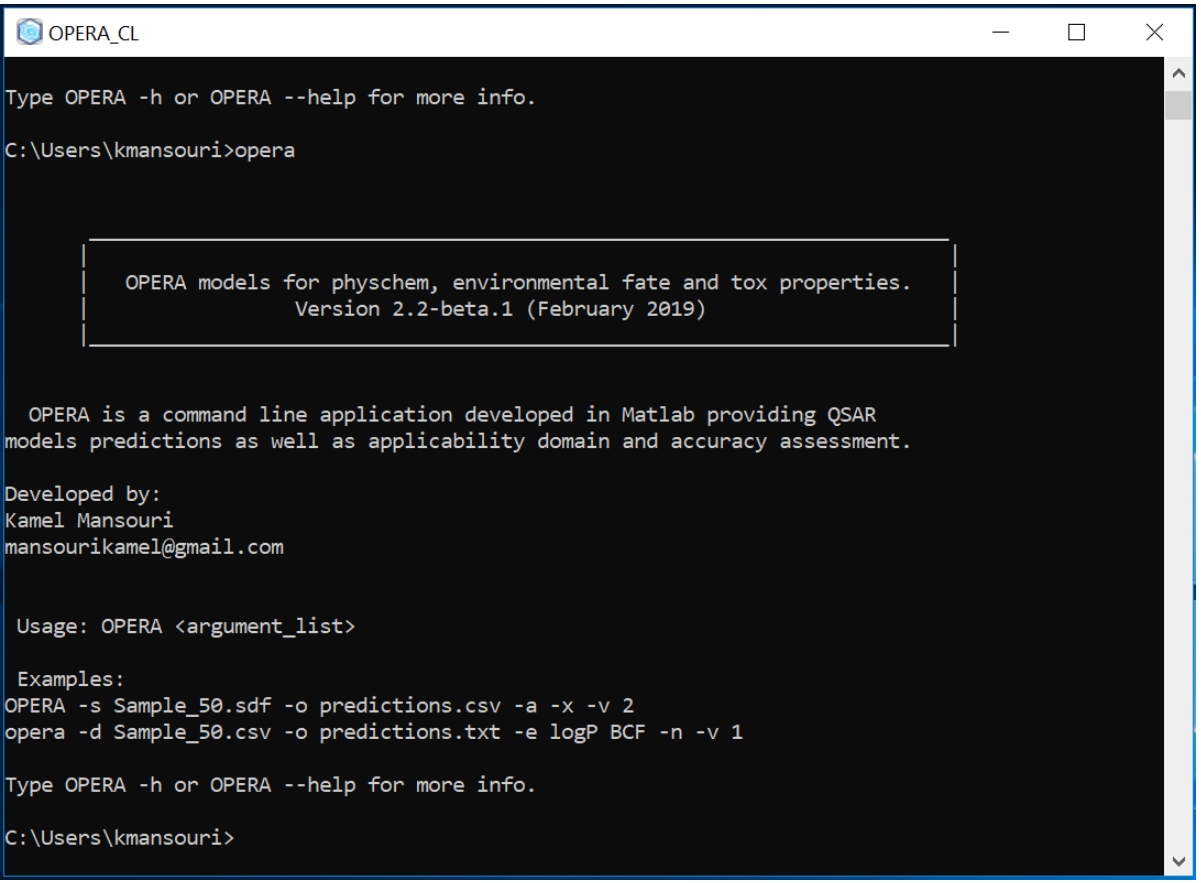
It will setup your environment variable and run the application in command line.

Double click the shortcut to run the app

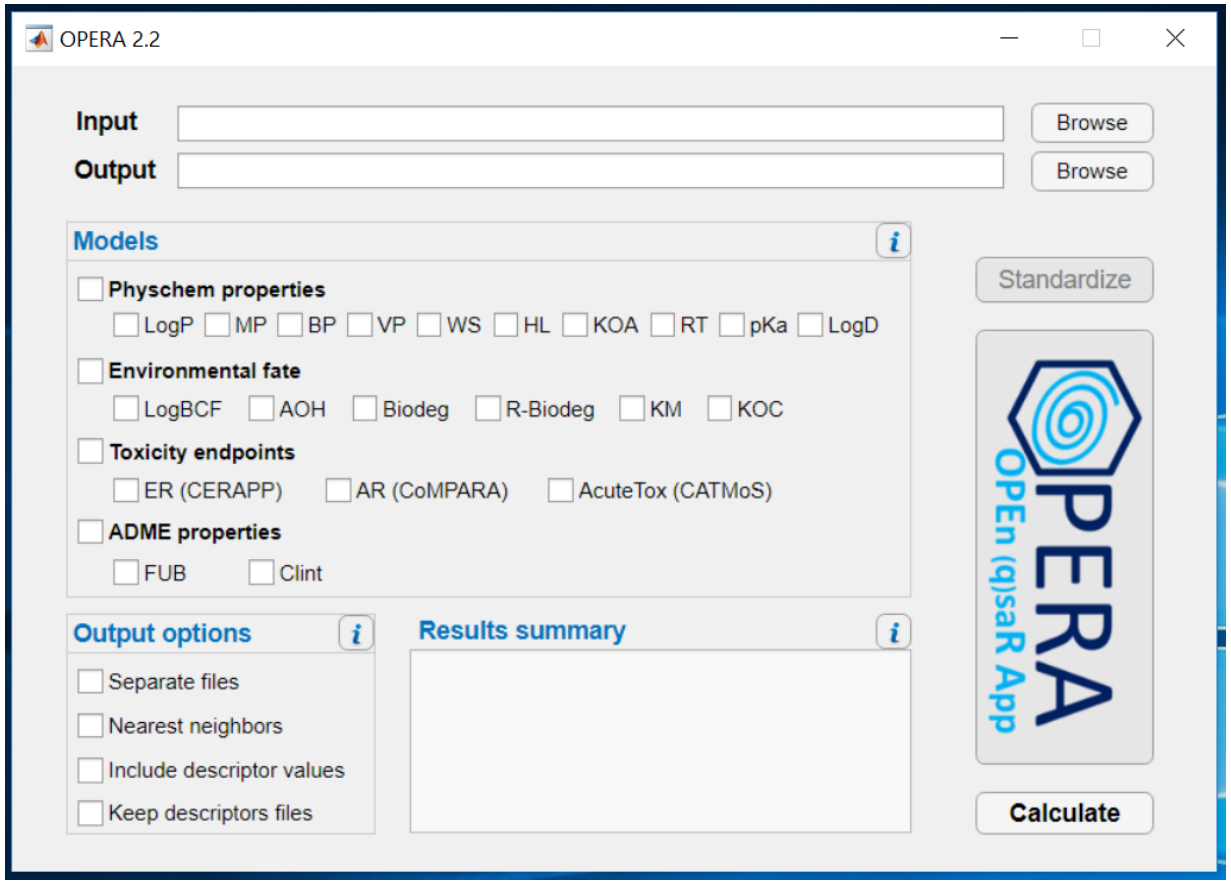
Command line

GUI

Ready to run the models



Command line



GUI

Accepted Input files are QSAR-ready structures in:

SMILES file:
- Extension .smi
- Tab delimited text file (structure \t ID)
- No headers & no empty lines

Sample_50.smi

```
NC(N)=N 50-01-1
CC1CC2C3CCC4=CC(=O)C=CC4(C)C3(F)C(O)CC2(C)C1(O)C(=O)CO
CC(=O)OCC(=O)C1(O)CCC2C3CCC4=CC(=O)CCC4(C)C3C(=O)CC12C
CCC1(c2ccccc2)C(=O)NC(=O)NC1=O 50-06-6
C[N+](CCOC(=O)C(O)(C1CCCCC1)c1ccccc1)(CC)CC 50-10-2
CN1C(=O)NC(=O)C(CC)(CC)C1=O 50-11-3
CCC1(NC(=O)N(C)C1=O)c1ccccc1 50-12-4
CC(C)C(C)C=CC(C)C1CCC2C(CCCC21C)=CC=C1CC(O)CCC1=C 50-
O=P1(NCCCO1)N(CCCl)CCCl 50-18-0
CCC(O)(COC(N)=O)c1ccccc1 50-19-1
CC(O)C(O)=O 50-21-5
CC12CCC(=O)C=C1CCC1C2C(O)CC2(C)C(CCC21)C(=O)CO 50-22-6
CC12CCC(=O)C=C1CCC1C2C(O)CC2(C)C1CCC2(O)C(=O)CO 50-23-7
CC12CCC3C(CCc4cc(O)ccc43)C1CC(O)C2O 50-27-1
CC12CCC3C(CCc4cc(O)ccc43)C1CCC2O 50-28-2
ClC(Cl)(Cl)C(c1ccc(Cl)cc1)c1ccc(Cl)cc1 50-29-3
OC(=O)c1c(Cl)cccc1Cl 50-30-6
OC(=O)c1c(Cl)c(Cl)ccc1Cl 50-31-7
c1c2ccccc2c2ccc3cccc4ccc1c2c34 50-32-8
CCCCC1C(=O)N(c2ccccc2)N(c2ccccc2)C1=O 50-33-9
C[N+](CCOC(=O)C1c2ccccc2Oc2ccccc21)(C(C)C)C(C)C 50-34-0
O=C1CCC(C(=O)N1)N1C(=O)c2ccccc2C1=O 50-35-1
COC(=O)C1C2CCC(CC1OC(=O)c1ccccc1)N2C 50-36-2
CN1CC(C=C2C1Cc1c[nH]c3cccc2c31)C(=O)N(CC)CC 50-37-3
CN1c2[n]c[n](C)c2C(=O)N(CC(C)O)C1=O 50-39-5
```

SDF/Mol file (v2000):
- Extension .sdf/.mol
- Different blocks
- Single or multiple molecules

Sample_50.smi | Sample_50.sdf

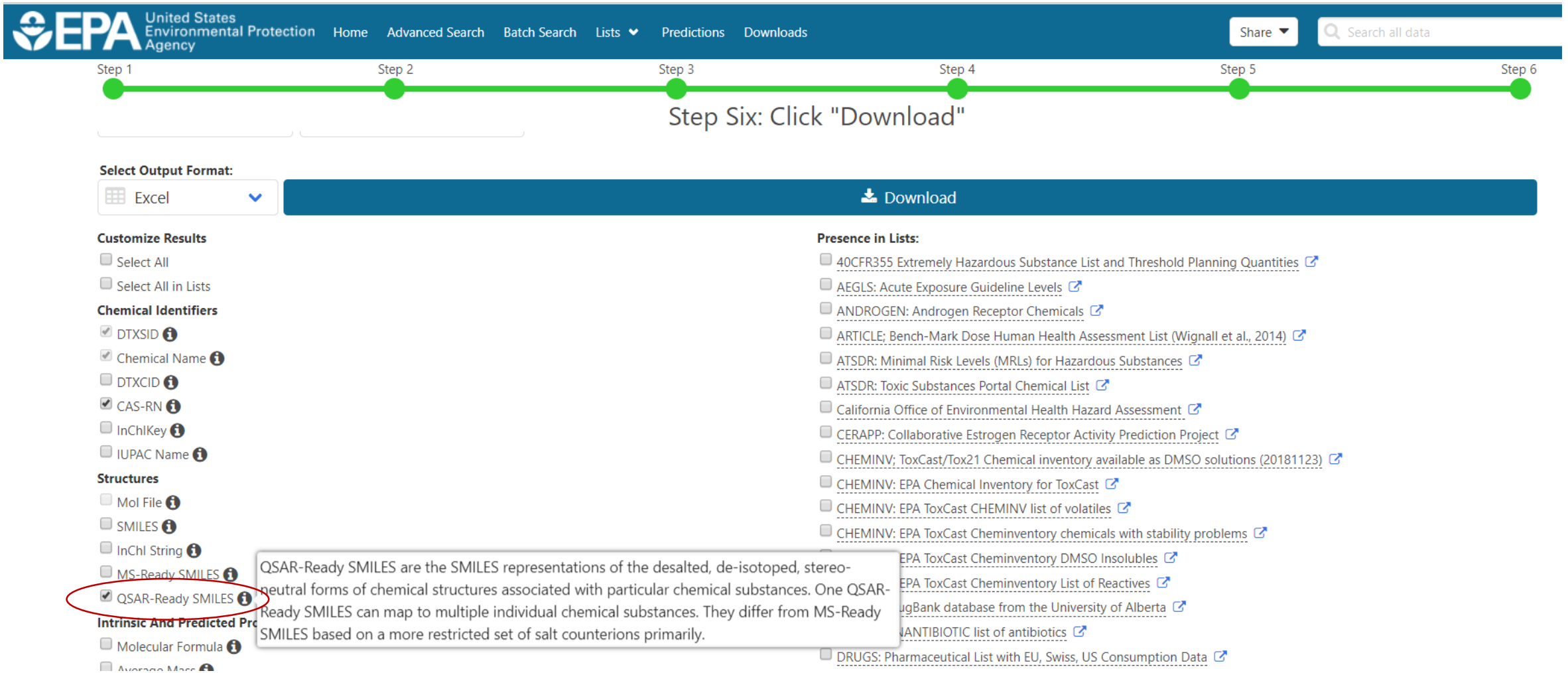
```
50-01-1
  -INDIGO-01081610012D

  9  8  0  0  0  0  0  0  0  0999 V2000
    0.0000    0.0000    0.0000 C   0  0
    1.6000    0.0000    0.0000 N   0  0
    2.4000    1.3856    0.0000 H   0  0
    2.4000   -1.3856    0.0000 H   0  0
   -0.8000   -1.3856    0.0000 N   0  0
   -2.4000   -1.3856    0.0000 H   0  0
    0.0000   -2.7713    0.0000 H   0  0
   -0.8000    1.3856    0.0000 N   0  0
   -2.4000    1.3856    0.0000 H   0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  2  4  1  0  0  0  0
  1  5  1  0  0  0  0
  5  6  1  0  0  0  0
  5  7  1  0  0  0  0
  1  8  2  0  0  0  0
  8  9  1  0  0  0  0
M  END
> <ChemID>
100002.0

> <CAS>
50-01-1
```

Descriptors files:
- Extension .csv
- Calculated by PaDEL and/or CDK
- Continuous and/or fingerprints

	A	B	C	D	E	F	G
	Name	nAcid	ALogP	ALogp2	AMR	apol	naAromAt
	100001	0	0	0	0	43.20152	20
	100002	0	0	0	0	48.0551	22
	100003	0	0	0	0	39.68152	18
	100004	0	0	0	0	14.56076	6
	100005	0	1.8631	3.471142	29.4773	19.8951	0
	100006	0	0.1248	0.015575	9.7308	27.78793	10
	100007	0	0	0	0	31.30793	14
	100008	0	0.4562	0.208118	4.601	29.54793	12
0	100009	0	0.642	0.412164	5.5021	26.02793	10
1	100010	0	0	0	0	22.93434	10
2	100011	0	0.642	0.412164	5.5021	26.02793	10
3	100012	0	0	0	0	27.78793	12
4	100013	0	0.789	0.622521	15.0732	21.17434	6
5	100014	0	0.5906	0.348808	7.7935	20.98076	9
6	100015	0	1.284	1.648656	11.0042	20.74793	6
7	100016	0	1.926	3.709476	16.5063	23.84152	6
8	100017	0	2.568	6.594624	22.0084	26.9351	6
9	100018	0	0.0686	0.004706	23.2621	18.56152	0
0	100019	0	0.5729	0.328214	14.2678	23.84152	6

QSAR-ready structures:

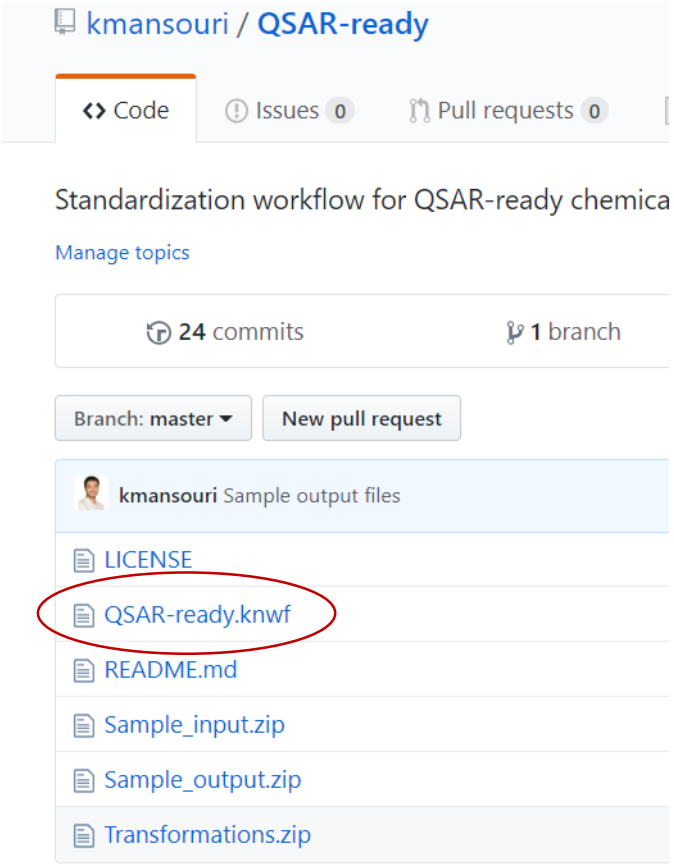
- Option 1: download the QSAR-ready SMILES from the EPA CompTox Dashboard: https://comptox.epa.gov/dashboard/dsstoxdb/batch_search

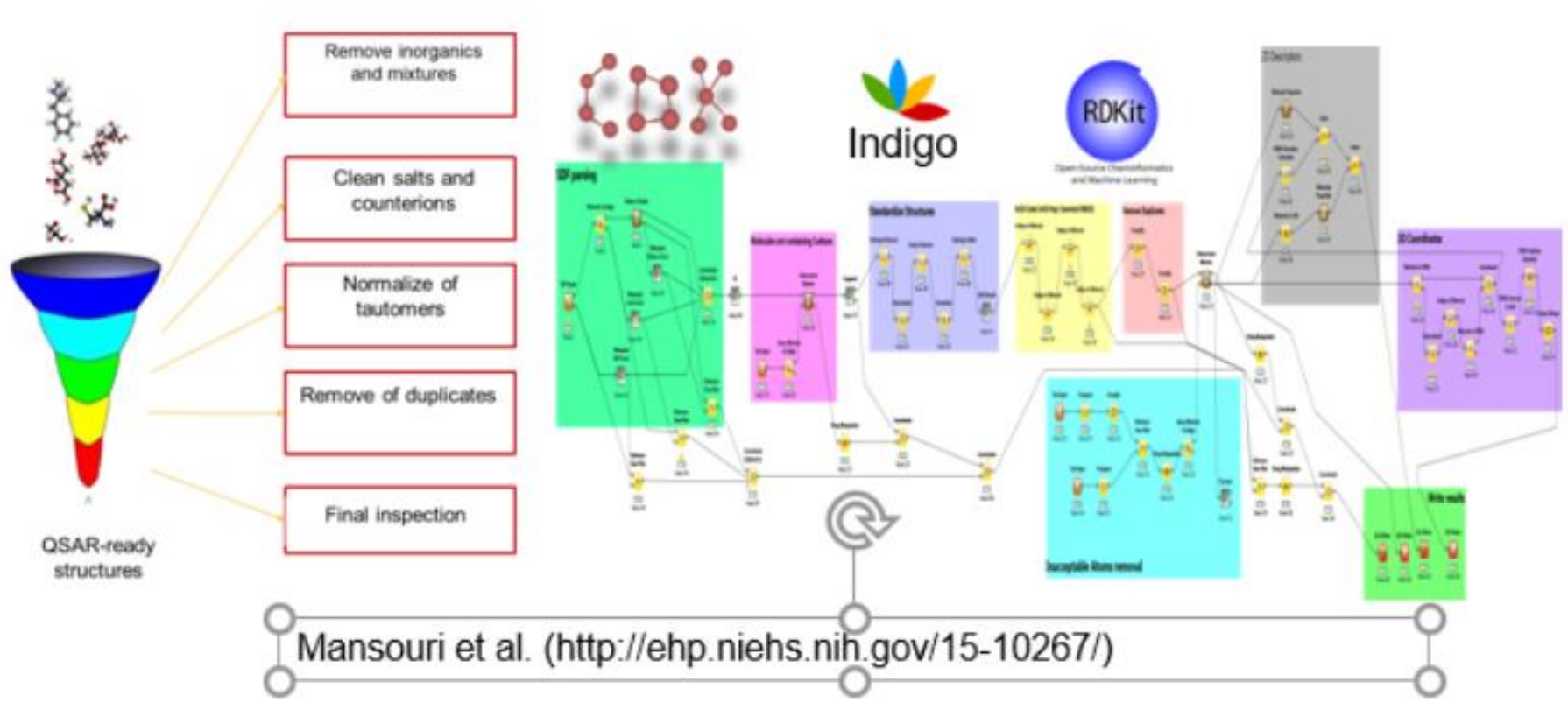


QSAR-ready structures:

- Option 2: run the KNIME QSAR-ready standardization workflow:

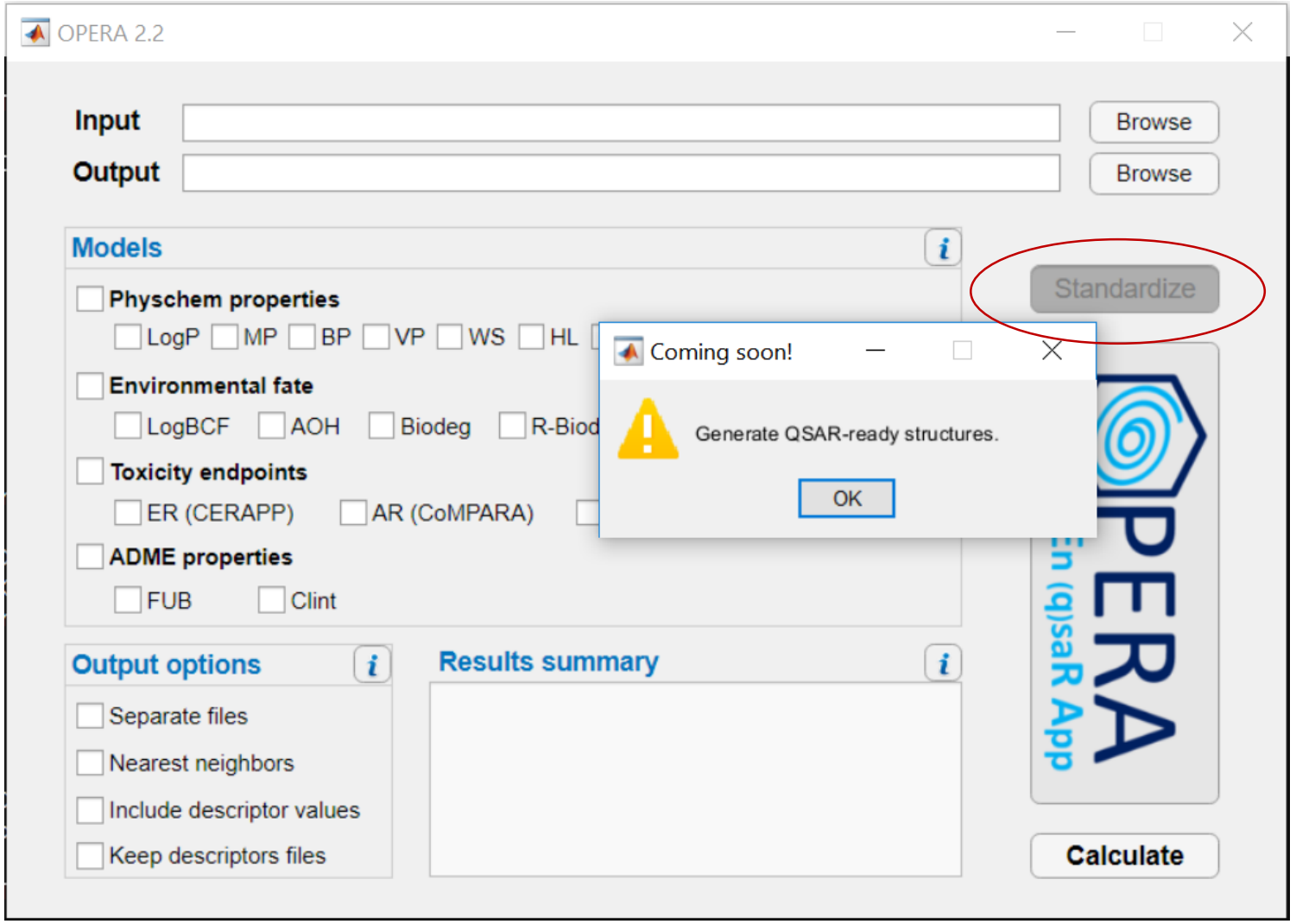
https://github.com/kmansouri/QSAR-ready



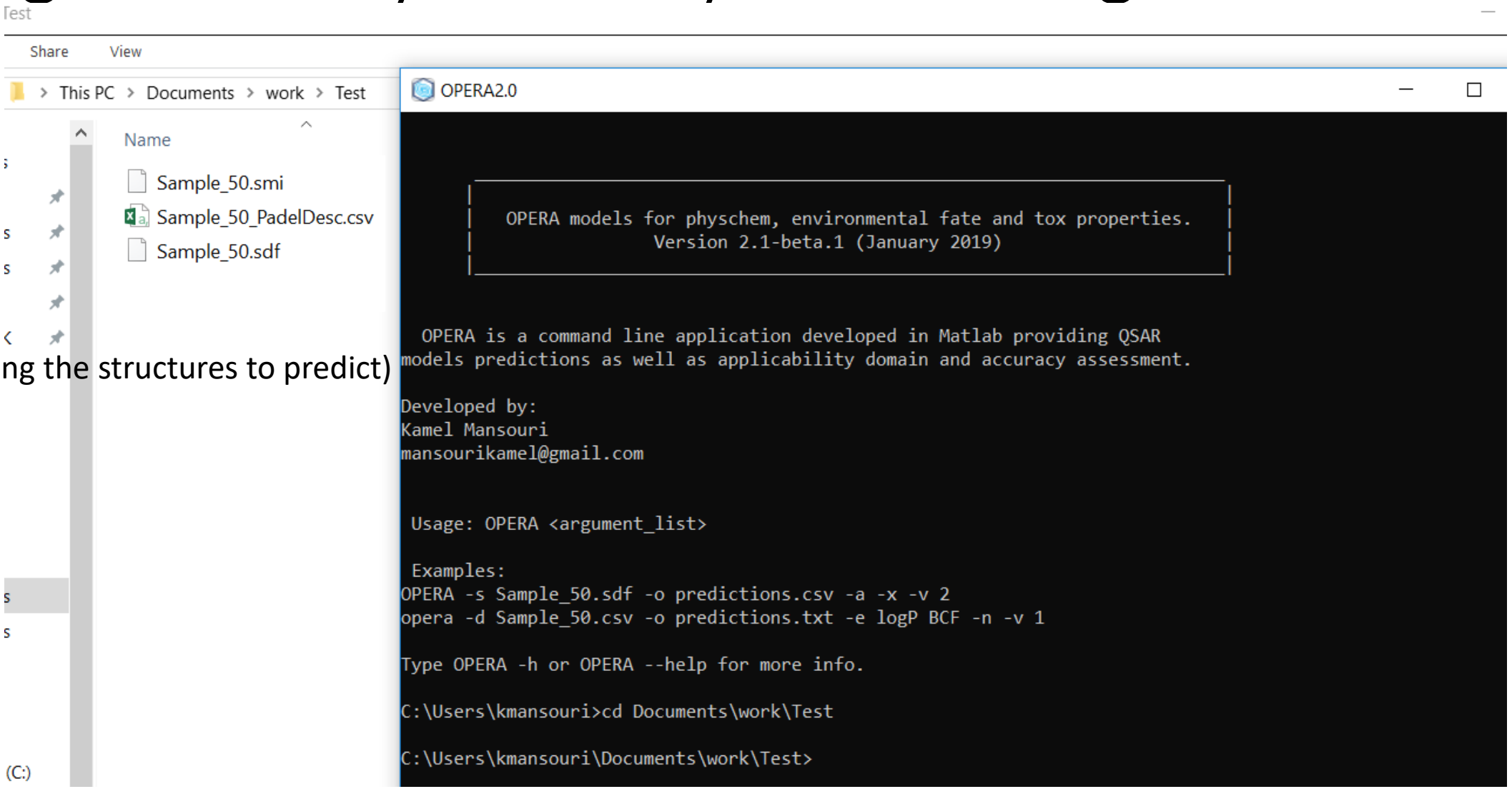


QSAR-ready structures:

- Option 3 (in the Future): Use the standardize option in OPERA



Command line:
Change directory “cd” to your working folder



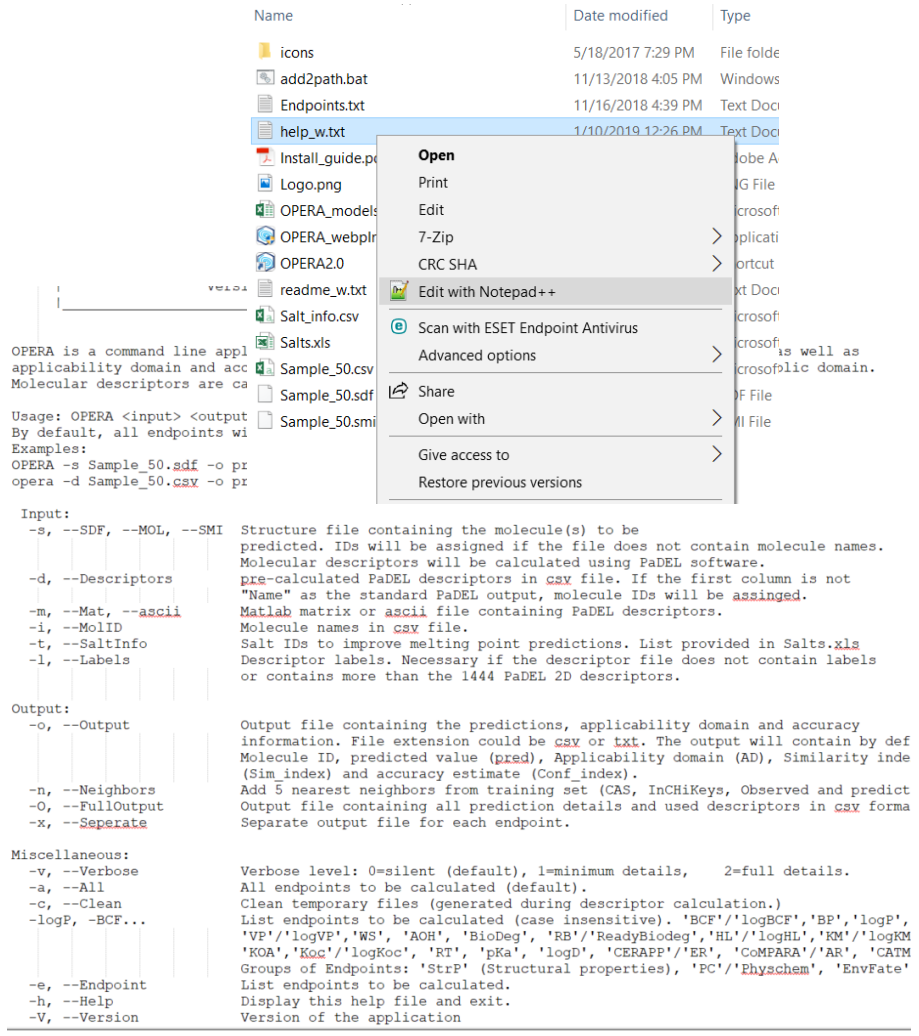
(Folder containing the structures to predict)

Use the commands in the help file or type: “opera –h” for help

```
C:\Users\kmansouri\Documents\work\Test>opera -h
```

OPERA2.0

```
 ____________________________________________________________
|                                                            |
|   OPERA models for physchem, environmental fate and tox properties.   |
|                Version 2.1-beta.1 (January 2019)           |
|____________________________________________________________|


OPERA is a command line application developed in Matlab providing QSAR models predictions as well as
applicability domain and accuracy assessment. All models are built on curated data from public domain.
Molecular descriptors are calculated using PaDEL and CDK software.

Usage: OPERA <input> <output> [Options]
By default, all endpoints will be calculated.
Examples:
OPERA -s Sample_50.sdf -o predictions.csv -a -x -n -v 2
opera -d Sample_50.csv -o predictions.txt -e logP BCF -v 1

 Input:
  -s, --SDF, --MOL, --SMI  Structure file containing the molecule(s) to be
                           predicted. IDs will be assigned if the file does not contain molecule names.
                           Molecular descriptors will be calculated using PaDEL software.
  -d, --Descriptors        pre-calculated PaDEL descriptors in csv file. If the first column is not
                           "Name" as the standard PaDEL output, molecule IDs will be assinged.
  -m, --Mat, --ascii       Matlab matrix or ascii file containing PaDEL descriptors.
  -i, --MolID              Molecule names in csv file.
  -t, --SaltInfo           Salt IDs to improve melting point predictions. List provided in Salts.xls
  -l, --Labels             Descriptor labels. Necessary if the descriptor file does not contain labels
                           or contains more than the 1444 PaDEL 2D descriptors.

Output:
  -o, --Output             Output file containing the predictions, applicability domain and accuracy
                           information. File extension could be csv or txt. The output will contain by default:
                           Molecule ID, predicted value (pred), Applicability domain (AD), Similarity index
                           (Sim_index) and accuracy estimate (Conf_index).
  -n, --Neighbors          Add 5 nearest neighbors from training set (CAS, InCHiKeys, Observed and predicted values)
  -O, --FullOutput         Output file containing all prediction details and used descriptors in csv format.
  -x, --Seperate           Separate output file for each endpoint.

Miscellaneous:
  -v, --Verbose            Verbose level: 0=silent (default), 1=minimum details,    2=full details.
  -a, --All                All endpoints to be calculated (default).
  -c, --Clean              Clean temporary files (generated during descriptor calculation.)
  -logP, -BCF...           List endpoints to be calculated (case insensitive). 'BCF'/'logBCF','BP','logP','MP',
                           'VP'/'logVP','WS', 'AOH', 'BioDeg', 'RB'/'ReadyBiodeg','HL'/'logHL','KM'/'logKM',
                           'KOA','Koc'/'logKoc', 'RT', 'pKa', 'logD', 'CERAPP'/'ER', 'CoMPARA'/'AR', 'CATMoS/AcuteTox'.
                                          Groups of Endpoints: 'StrP' (Structural properties), 'PC'/'Physchem', 'EnvFate'/'EF', 'Tox' (ER, AR, AcuteTox).
  -e, --Endpoint               List endpoints to be calculated.
  -h, --Help               Display this help file and exit.
  -V, --Version            Version of the application
```



Examples:

```
Usage: OPERA <argument_list>
```

General command: “OPERA <input> <output> <options> “ no specific order for the arguments.

```
>opera -s Sample_50.sdf -o predictions.csv -logP -v 1
```

Using the sample sdf file (50 chemicals) to calculate logP only in minimum verbose mode (v=1). Endpoints names (only) are case insensitive.

```
>opera -s Sample_50.sdf -o predictions.csv
```

Simplest command specifying only the input and output. By default, OPERA will calculate all endpoints and output them to the same file in silent mode (V=0 by default).

```
>opera -d PadelDesc.csv -o predictions.csv -LOGP -BCF -v 2
```

Running one of the previous commands will generate the descriptor file “PadelDesc.csv” for this list of chemicals that you can use to run other models.

```
>opera -d PadelDesc.csv -o predictions.csv -physchem -v 2 -s Sample_50.sdf
```

This runs a list of models at the same time without running all of them. “physchem” or “pc” will run 10 models.

```
>opera -s Sample_50.smi -o predictions.csv -CATMOS -v 1
```

This will run CATMoS models only using “CATMoS” or “AcuteTox” (case insensitive).

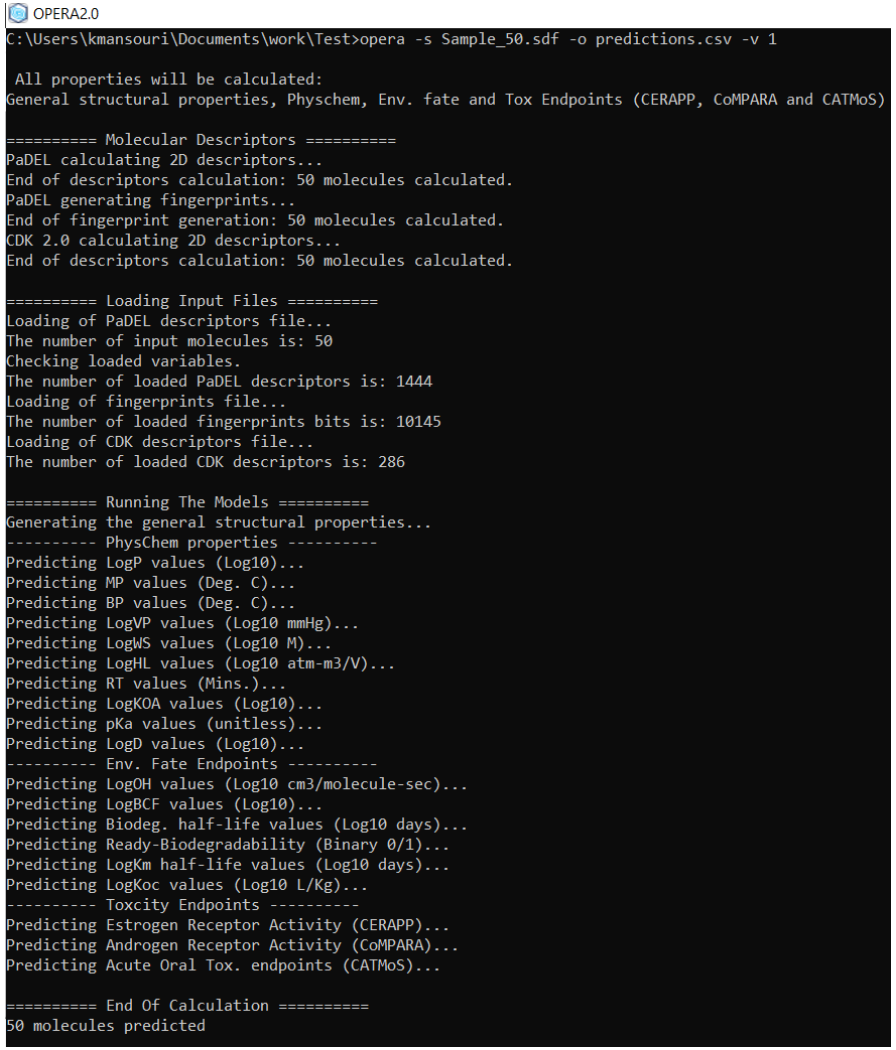
```
>opera -s Sample_50.sdf -o predictions.csv -all -n -x -v 1 -c
```

This will run all models. “-n” to get the nearest neighbors and their experimental and predicted data. “-x” will output each one of them in a separate file. “-c” or”-clean” to delete generated descriptors afterwards.

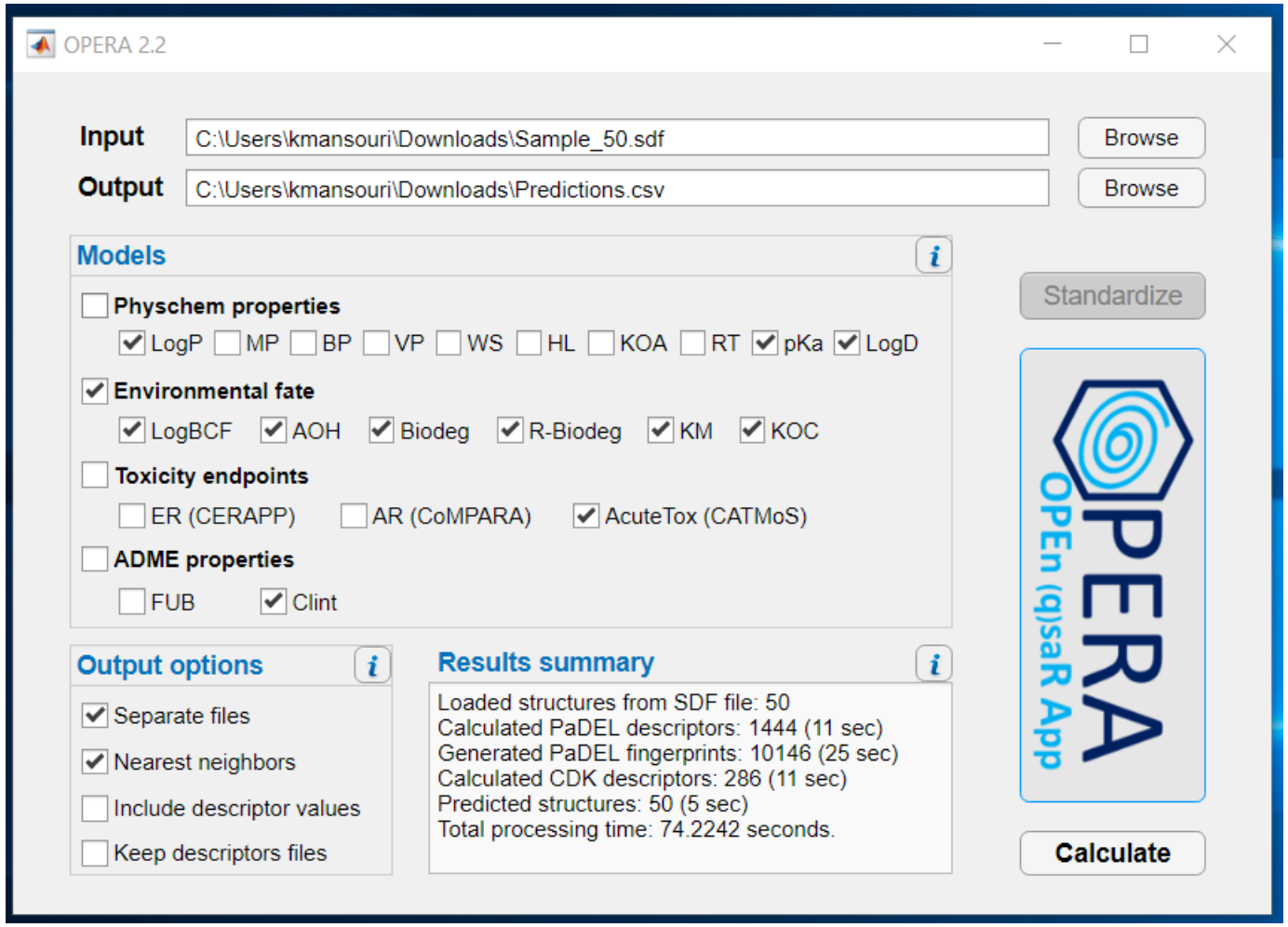
```
>opera -s Sample_50.sdf -d PadelDesc.csv -o predictions.csv -Tox -n -x -v 1
```

This will run all toxicity endpoints including CERAPP (ER), CoMPARA (AR) and CATMoS (AcuteTox).

Running the provided sample structures



Command line



GUI

Sample output:

	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1	LogP_prec	AD_LogP	AD_index	Conf_inde	MP_pred	AD_MP	AD_index	Conf_inde	BP_pred	AD_BP	AD_index	Conf_inde
2	-1.16675	1	0.480419	0.514066	56.96995	0	0.845536	0.591406	257.4774	0	0.148019	0.459055
3	1.865338	1	0.900462	0.860512	240.0609	1	0.886896	0.77831	391.2263	0	0.156367	0.289241
4	2.220271	1	0.908599	0.854532	225.4009	1	0.868502	0.769243	418.5021	0	0.198701	0.29612
5	1.418476	1	0.888307	0.832005	178.3617	1	0.907997	0.792323	255.0721	0	0.102487	0.429512
6	3.263077	1	0.475522	0.511194	113.4288	1	0.423574	0.396192	303.5064	1	0.218644	0.377263
7	1.250174	1	0.88672	0.826925	140.6504	1	0.870612	0.678357	244.3635	0	0.199114	0.408801
8	1.661618	1	0.884391	0.880548	140.7473	1	0.869315	0.704217	313.0569	0	0.227134	0.451173
9	8.521042	0	0.423516	0.451172	109.0241	1	0.918952	0.808039	388.1919	0	0.288908	0.478521
10	0.794911	1	0.916139	0.831594	53.26655	1	0.889102	0.805115	300.1238	0	0.215153	0.397557
11	-1.34435	1	0.542563	0.511082	70.8003	1	0.862021	0.603289	300.3755	0	0.275189	0.513219
12	-0.76893	1	0.880398	0.785205	45.61057	1	0.87657	0.805093	208.6963	1	0.419094	0.402072
13	2.075537	1	0.885388	0.774448	190.2265	1	0.890244	0.776764	400.3502	0	0.227297	0.322959
14	1.653221	1	0.887992	0.897769	221.5952	1	0.91094	0.869202	389.2867	0	0.161782	0.29391
15	2.635143	1	0.863338	0.699103	270.1702	1	0.856718	0.608162	382.0755	0	0.189731	0.315207
16	3.907586	1	0.87718	0.772113	196.6646	1	0.881577	0.794906	372.7094	1	0.273881	0.367613
17	6.91244	1	0.871424	0.810684	104.6093	1	0.899273	0.844186	350.3782	1	0.549401	0.606396
18	2.64288	1	0.919871	0.754499	157.4414	1	0.94492	0.84299	293.8553	1	0.729356	0.715913
19	3.536705	1	0.611748	0.663621	147.8905	1	0.94981	0.796142	300.6033	1	0.552324	0.69184
20	6.128521	1	0.947304	0.88721	223.6441	1	0.978058	0.708651	485.9157	0	0.937575	0.871229

predictions

In this example, all endpoints were predicted and the output is written in a single file “predictions.csv”.

For each model, the predicted value is associated with applicability domain and accuracy estimates.

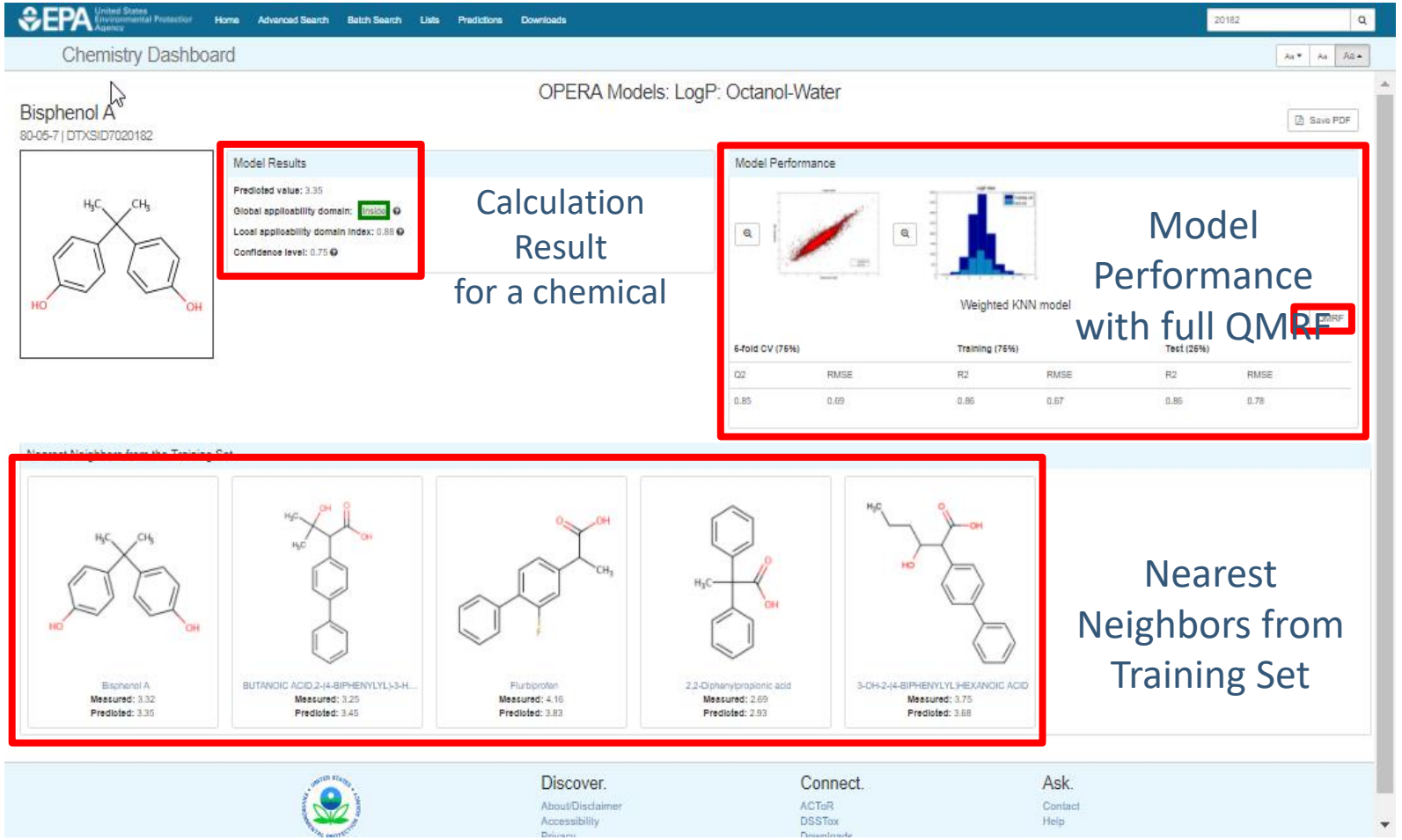
AD_logP (0/1): Global applicability domain considering the whole chemical space of the model.

AD_index [0-1]: Local applicability domain based on the similarity to the 5 nearest neighbors

Conf_index [0-1]: Accuracy estimate based on the predictions of the 5 nearest neighbors.

OPERA prediction report on the EPA Dashboard

https://comptox.epa.gov/dashboard



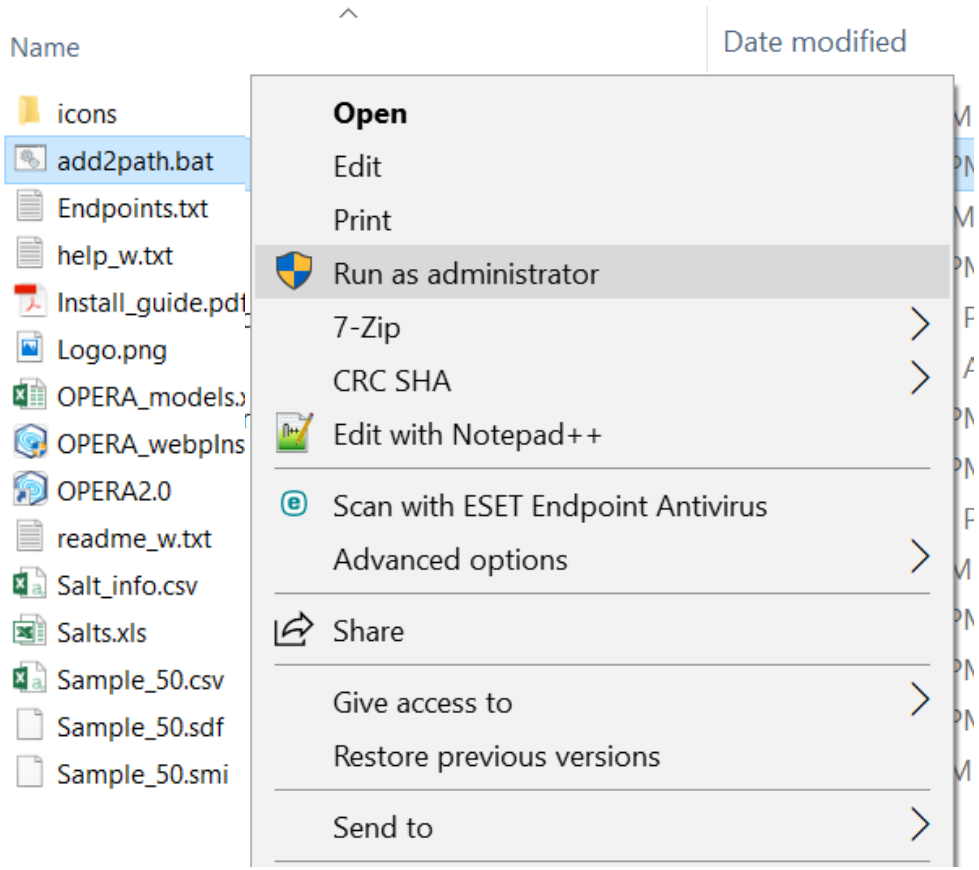
More details in the QMRF reports available on the EPA dashboard, the Github repo and the JRC QSAR database

Optional and only for expert users of command line:

Run add2path.bat as admin to permanently edit system %PATH%.

Only needed to install for all users or to use in a pipeline/ workflow.

Run only ONCE to avoid truncated path!



Thank you!

Register on the GitHub repo for the updates
https://github.com/NIEHS/OPERA

References:

[1] Mansouri K. et al. J Cheminform (2018) https://doi.org/10.1186/s13321-018-0263-1.

[2] Mansouri, K. et al. SAR and QSAR in Env. Res. (2016). https://doi.org/10.1080/1062936X.2016.1253611

[3] Williams A. J. et al. J Cheminform (2017) https://doi.org/10.1186/s13321-017-0247-6

[4] The CompTox Chemistry Dashboard (https://comptox.epa.gov/dashboard)

[5] JRC QSAR Model Database https://qsardb.jrc.ec.europa.eu/qmrf/endpoint

Your feedback will be appreciated!

Kamel.mansouri@nih.gov
kmansouri@ils-inc.com
mansourikamel@gmail.com