Cheminformatics Machine Learning for Homogeneous Catalysis

Schrödinger

Materials

Cheminformatics Machine Learning for Homogeneous Catalysis

Topics: Catalysis and Reactive Systems, Semiconductors, Energy Capture and Storage, Metals Alloys and Ceramics, Informatics and Team Collaboration

Methodology: Machine Learning

Products Used: MS Maestro, AutoQSAR, MS Informatics

This tutorial is written for use with a 3-button mouse with a scroll wheel.

Words found in the **Glossary of Terms** are shown like this: Workspace

Abstract:

In this tutorial, we will learn to develop and use a machine learning model to predict reaction rate constants for iridium catalysts.

Tutorial Content

1. Introduction

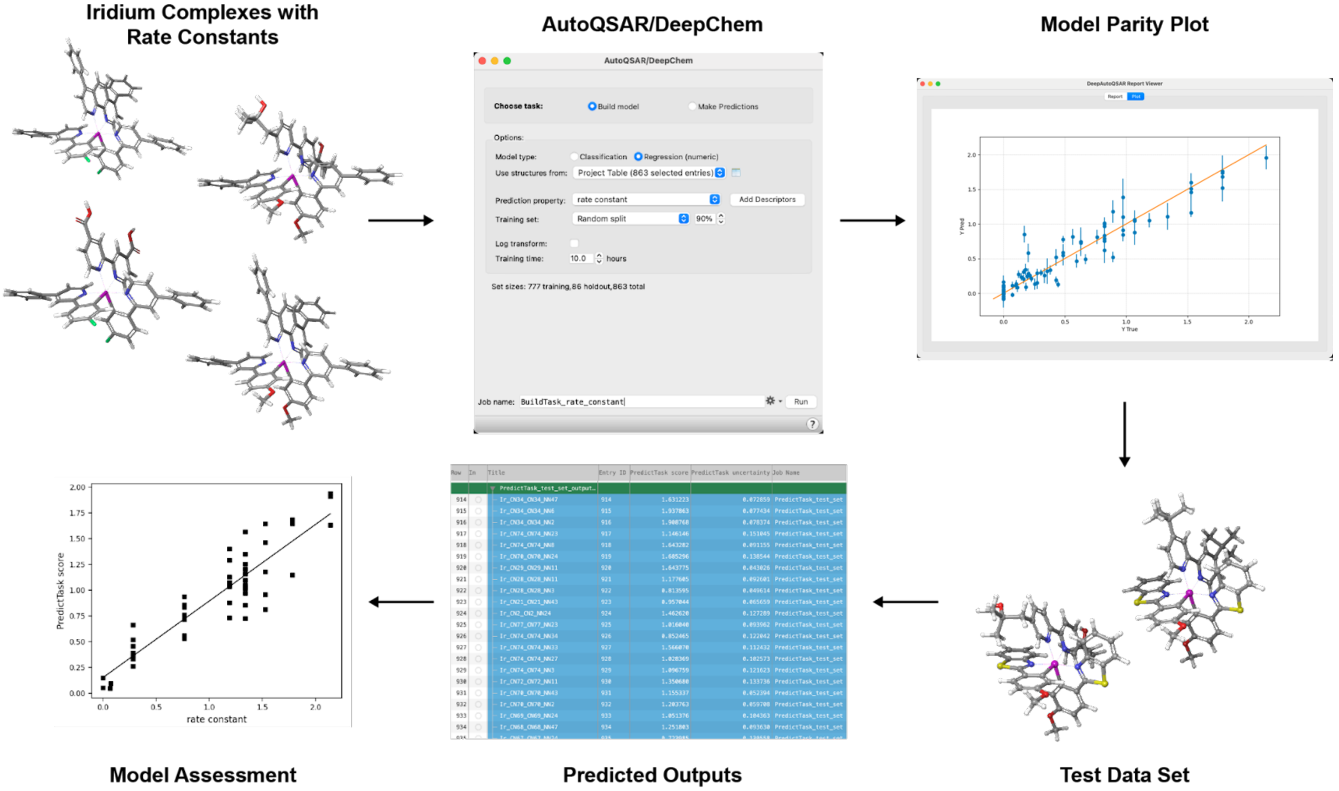
Discovering new catalysts for improved reactivity or selectivity is challenging because of the large number of laborious experiments or stepwise quantum mechanical calculations necessary to explore the catalyst design space. Alternative to these approaches, employing machine learning (ML) for catalyst discovery and design is a promising avenue to rapidly screen catalysts for enhanced properties (see **References** for recent literature examples).

A useful ML tool is Quantitative Structure-Activity Relationships (QSAR), which can efficiently predict material properties for a wide-range of molecules. Schrödinger's AutoQSAR tools automates the generation of accurate QSAR models, which allows users to leverage machine learning tools without extensive background knowledge. For a complete description of how AutoQSAR automatically tests various models and makes selections, visit the Machine Learning for Materials Science tutorial.

DeepAutoQSAR integrates graph convolutional neural networks into the traditional AutoQSAR workflow, where DeepAutoQSAR treats a molecule as a graph consisting of nodes as atoms and edges as bonds. DeepAutoQSAR has been found to outperform traditional AutoQSAR for 'large' datasets (>5000 molecules) and perform similarly to traditional AutoQSAR for 'small' datasets (<5000 molecules) (see comparison here). A distinct advantage of DeepAutoQSAR is its ability to identify hidden patterns relevant to the property of interest through a series of convolution operations. You can read more about DeepAutoQSAR on our webpage as well as the references therein.

In this tutorial, we will use the DeepAutoQSAR panel in MS Maestro to create a machine learning model to predict rate constants for a radical reaction (reductive dehalogenation of aryl halide) catalyzed by a series of organometallic iridium complexes. The experimental data set is provided from a recent publication from Mdluhi *et al.* (High-throughput Synthesis and Screening of Iridium (III) Photocatalysts for the Fast and Chemoselective Dehalogenation of Aryl Bromides. DOI:10.1021/acscatal.0c02247). This experimental dataset explores a series of ~1000 $[Ir(C^\wedge N)_2(N^\wedge N)]^+$ photocatalysts (octahedral iridium complexes with three, bidentate ligands) and measures rate constants using high-throughput colorimetric monitoring.

Herein, we use a data set of 863 of the iridium complexes and the experimental rate constants to train and evaluate machine learning models. The DeepAutoQSAR panel is used to generate a model to predict rate constants by training on the structure of each Ir complex and the associated rate constant. To test the generalizability of the model, rate constants are predicted for an unseen set of 50 complexes. The overall workflow is summarized in *Figure 1*.



AutoQSAR/DeepChem

Choose task: ● Build model ○ Make Predictions

Options:

Model type: ○ Classification ● Regression (numeric)

Use structures from: Project Table (863 selected entries)

Prediction property: rate constant

Add Descriptors

Training set: Random split 90%

Log transform: □

Training time: 10.0 hours

Set sizes: 777 training,86 holdout,863 total

Job name: BuildTask_rate_constant

Run

Row	In	Title	Entry ID	PredictTask score	PredictTask uncertainty	Job Name
		PredictTask_test_set_output…				
914	○	Ir_CN34_CN34_NN47	914	1.631223	0.072859	PredictTask_test_set
915	○	Ir_CN34_CN34_NN6	915	1.937863	0.077434	PredictTask_test_set
916	○	Ir_CN34_CN34_NN2	916	1.908768	0.078374	PredictTask_test_set
917	○	Ir_CN74_CN74_NN23	917	1.146146	0.151045	PredictTask_test_set
918	○	Ir_CN74_CN74_NN8	918	1.643282	0.091155	PredictTask_test_set
919	○	Ir_CN70_CN70_NN24	919	1.685296	0.138544	PredictTask_test_set
920	○	Ir_CN29_CN29_NN11	920	1.643775	0.043026	PredictTask_test_set
921	○	Ir_CN28_CN28_NN11	921	1.177605	0.092601	PredictTask_test_set
922	○	Ir_CN28_CN28_NN3	922	0.813595	0.049614	PredictTask_test_set
923	○	Ir_CN21_CN21_NN43	923	0.957044	0.065659	PredictTask_test_set
924	○	Ir_CN2_CN2_NN24	924	1.462620	0.127289	PredictTask_test_set
925	○	Ir_CN77_CN77_NN23	925	1.016040	0.093962	PredictTask_test_set
926	○	Ir_CN74_CN74_NN34	926	0.852465	0.122042	PredictTask_test_set
927	○	Ir_CN74_CN74_NN33	927	1.566070	0.112432	PredictTask_test_set
928	○	Ir_CN74_CN74_NN27	928	1.028369	0.102573	PredictTask_test_set
929	○	Ir_CN74_CN74_NN3	929	1.096759	0.121623	PredictTask_test_set
930	○	Ir_CN72_CN72_NN11	930	1.350680	0.133736	PredictTask_test_set
931	○	Ir_CN70_CN70_NN43	931	1.155337	0.052394	PredictTask_test_set
932	○	Ir_CN70_CN70_NN2	932	1.203763	0.059708	PredictTask_test_set
933	○	Ir_CN69_CN69_NN24	933	1.051376	0.104363	PredictTask_test_set
934	○	Ir_CN68_CN68_NN47	934	1.251803	0.093630	PredictTask_test_set

Figure 1. Tutorial workflow showing the input Ir complexes, DeepAutoQSAR panel used to build machine learning models, and the output parity plot after model training. After training the model, an unseen test set was used to evaluate model performance. The workflow subsequently shows the Ir complexes, output predictions, and parity plot for the test set.

Note that while this dataset is small enough that AutoQSAR could also be used, this tutorial focuses on using DeepAutoQSAR, which produces slightly more accurate predictions than traditional AutoQSAR.

For additional practice with the DeepAutoQSAR workflow, but with a categorical classification task, see the Machine Learning for Sweetness tutorial.

For additional practice with AutoQSAR, tutorials are available using the Materials Science Maestro suite to predict properties of small molecules, polymers and periodic systems: Machine Learning for Materials Science, Polymer Descriptors for Machine Learning and Periodic Descriptors for Inorganic Solids.

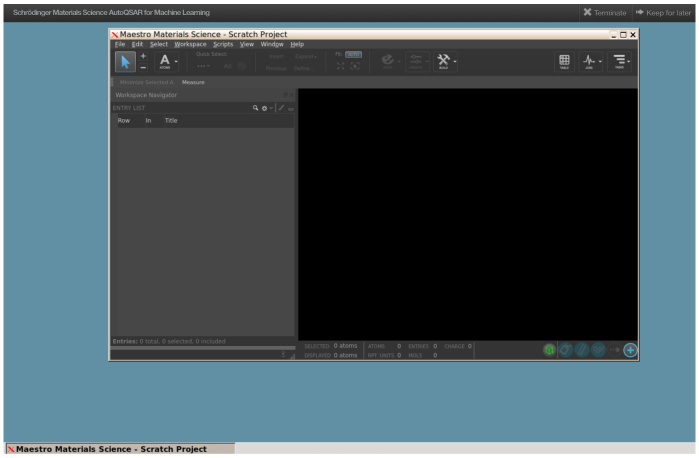
To learn about using pre-built machine learning models to predict volatility of organometallic complexes, please refer to the Machine Learning Property Prediction tutorial.

For alternative computational approaches for catalyst discovery, namely elucidating reaction mechanisms via various workflows, visit the Locating Transition States: Part 1 and Part 2 tutorials, as well as the Reaction Workflow for Polyethylene Insertion tutorial.

2. Creating Projects and Importing Structures

At the start of the session, change the file path to your chosen Working Directory in MS Maestro to make file navigation easier. Each session in MS Maestro begins with a default Scratch Project, which is not saved. A MS Maestro project stores all your data and has a `.prj` extension. A project may contain numerous entries corresponding to imported structures, as well as the output of modeling-related tasks. Once a project is saved, the project is automatically saved each time a change is made.

Structures can be built in MS Maestro or can be imported using **File > Import Structures** (or drag-and-dropped), and are added to the Entry List and Project Table. The Entry List is located to the left of the Workspace. The Project Table can be accessed by **Ctrl+T (Cmd+T)** or **Window > Project Table** if you would like to see an expanded view of your project data.

 Schrödinger Materials Science AutoQSAR for Machine Learning Terminate Keep for later Maestro Materials Science - Scratch Project File Edit Select Workspace Scripts View Window Help Measure Workspace Navigator ENTRY LIST Row In Title Entries: 0 total, 0 selected, 0 included SELECTED 0 atoms DISPLAYED 0 atoms Maestro Materials Science - Scratch Project	1. Launch the tool in the nanoHUB interface ○ Launching the tool will automatically open up MS Maestro

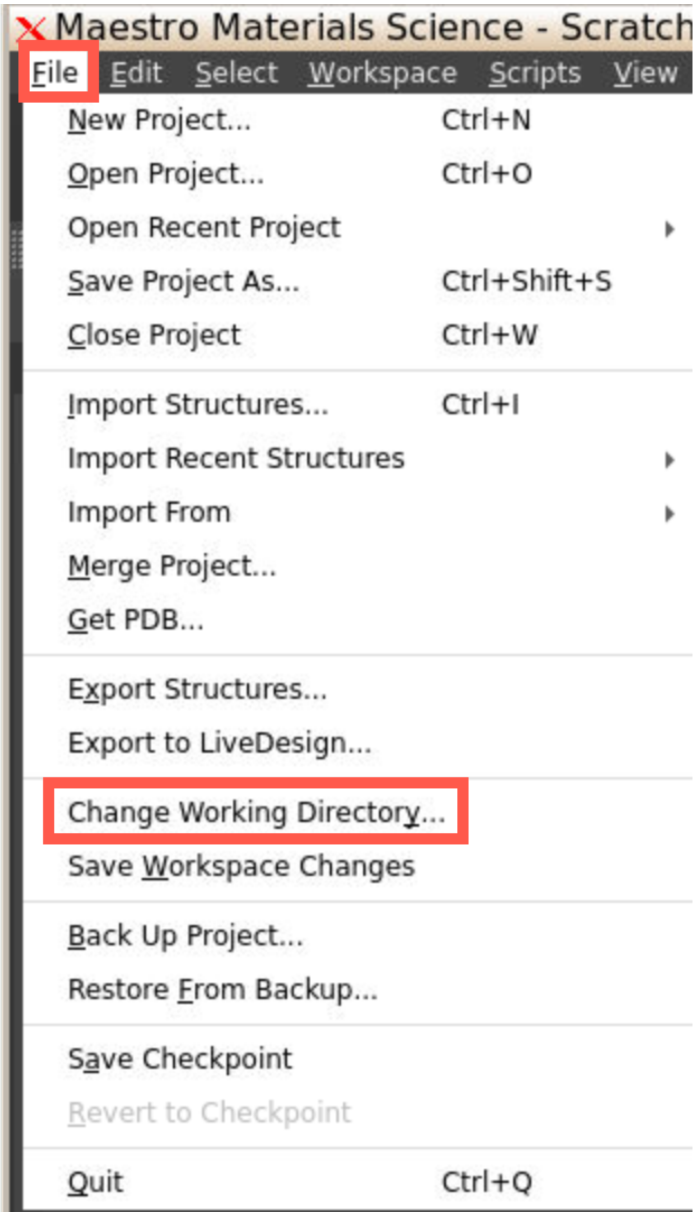
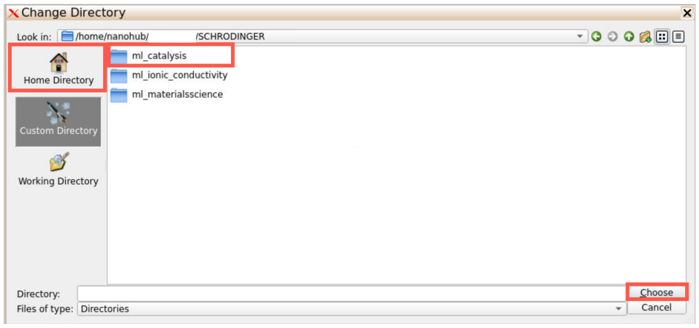


Figure 2-1. Change Working Directory option.

2. Go to **File > Change Working Directory**



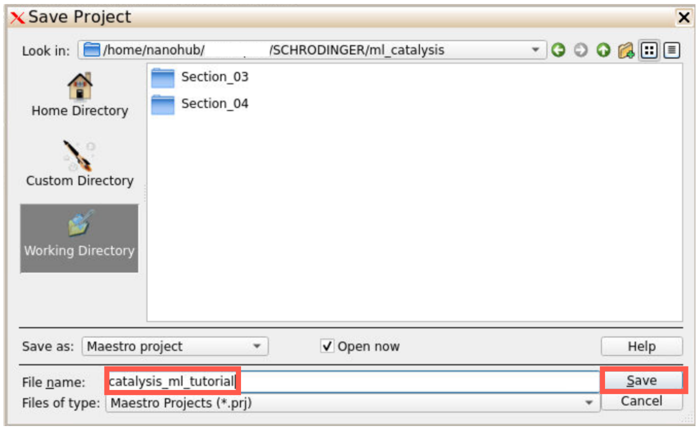
Look in: /home/nanohub/ /SCHRODINGER

Directory:

Files of type: Directories

Figure 2-2. Selecting the Working Directory.

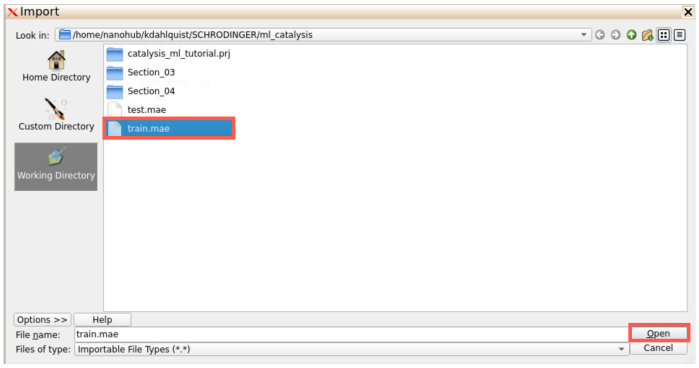
1. Navigate to your Home Directory then the SCHRODINGER directory
2. Select *catalysis_ml*, and click **Choose**
   - All files needed to execute this tutorial are included in this directory



Look in: /home/nanohub/ /SCHRODINGER/ml_catalysis
Save as: Maestro project ☑ Open now
File name: catalysis_ml_tutorial
Files of type: Maestro Projects (*.prj)

Figure 2-3. Save Project panel.

3. Go to **File > Save Project As**
4. Change the *File name* to **catalysis_ml_tutorial**, click **Save**
    - The project is now named `catalysis_ml_tutorial.prj`

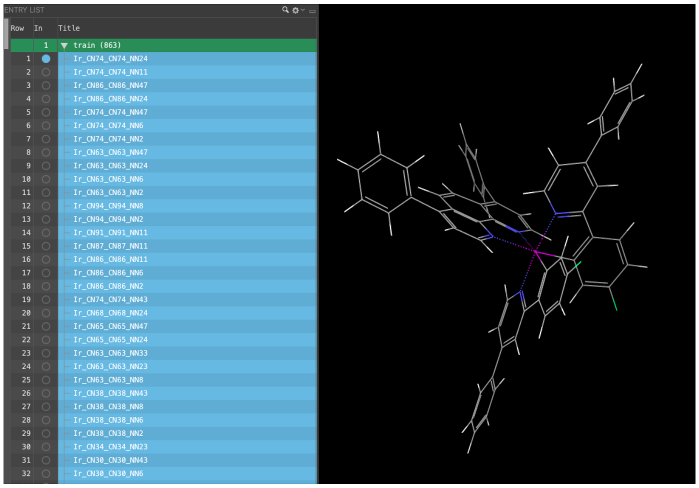


Look in: /home/nanohub/kdahlquist/SCHRODINGER/ml_catalysis
File name: train.mae
Files of type: Importable File Types (*.*)

Figure 2-4. Import the starting structures.

Let's import the data set:

5. Go to **File > Import Structures**
6. Navigate to where you downloaded the provided tutorial files (presumably in your working directory) and **choose** `train.mae` from the provided tutorial files
7. Click **Open**

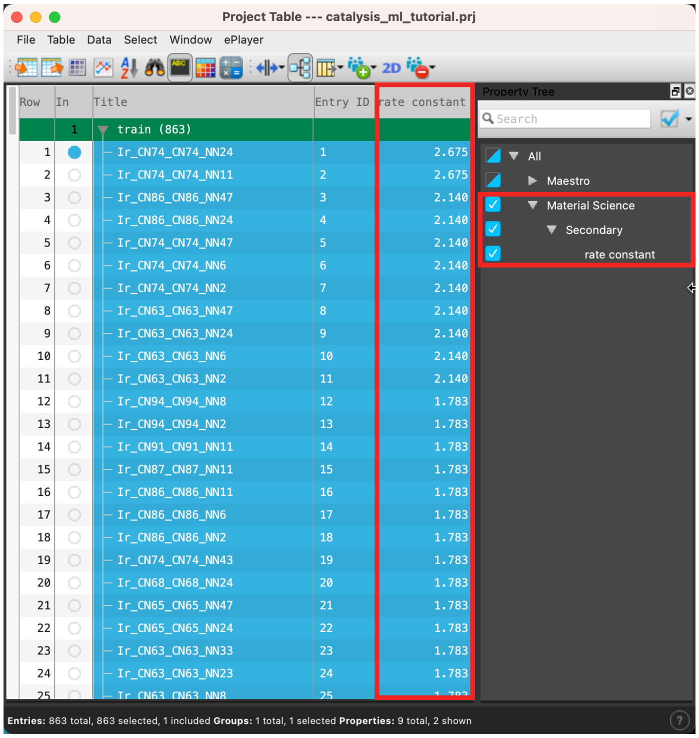


Row	In	Title
	1	train (863)
1	●	Ir_CN74_CN74_NN24
2	○	Ir_CN74_CN74_NN11
3	○	Ir_CN86_CN86_NN47
4	○	Ir_CN86_CN86_NN24
5	○	Ir_CN74_CN74_NN47
6	○	Ir_CN74_CN74_NN6
7	○	Ir_CN74_CN74_NN2
8	○	Ir_CN63_CN63_NN47
9	○	Ir_CN63_CN63_NN24
10	○	Ir_CN63_CN63_NN6
11	○	Ir_CN63_CN63_NN2
12	○	Ir_CN94_CN94_NN8
13	○	Ir_CN94_CN94_NN2
14	○	Ir_CN91_CN91_NN11
15	○	Ir_CN87_CN87_NN11
16	○	Ir_CN86_CN86_NN11
17	○	Ir_CN86_CN86_NN6
18	○	Ir_CN86_CN86_NN2
19	○	Ir_CN74_CN74_NN43
20	○	Ir_CN68_CN68_NN24
21	○	Ir_CN65_CN65_NN47
22	○	Ir_CN65_CN65_NN24
23	○	Ir_CN63_CN63_NN33
24	○	Ir_CN63_CN63_NN23
25	○	Ir_CN63_CN63_NN8
26	○	Ir_CN38_CN38_NN43
27	○	Ir_CN38_CN38_NN8
28	○	Ir_CN38_CN38_NN6
29	○	Ir_CN38_CN38_NN2
30	○	Ir_CN34_CN34_NN23
31	○	Ir_CN30_CN30_NN43
32	○	Ir_CN30_CN30_NN6

Figure 2-5. The entry list and a complex after importing.

The entry list is updated to include the 863 entries. Feel free to stylize and visualize any of the provided structures.

Note: The model complexes were prepared using Materials Science Maestro structure building capabilities (see the Organometallic Complexes tutorial for relevant workflows).



Row	In	Title	Entry ID	rate constant
	1	train (863)		
1	●	Ir_CN74_CN74_NN24	1	2.675
2	○	Ir_CN74_CN74_NN11	2	2.675
3	○	Ir_CN86_CN86_NN47	3	2.140
4	○	Ir_CN86_CN86_NN24	4	2.140
5	○	Ir_CN74_CN74_NN47	5	2.140
6	○	Ir_CN74_CN74_NN6	6	2.140
7	○	Ir_CN74_CN74_NN2	7	2.140
8	○	Ir_CN63_CN63_NN47	8	2.140
9	○	Ir_CN63_CN63_NN24	9	2.140
10	○	Ir_CN63_CN63_NN6	10	2.140
11	○	Ir_CN63_CN63_NN2	11	2.140
12	○	Ir_CN94_CN94_NN8	12	1.783
13	○	Ir_CN94_CN94_NN2	13	1.783
14	○	Ir_CN91_CN91_NN11	14	1.783
15	○	Ir_CN87_CN87_NN11	15	1.783
16	○	Ir_CN86_CN86_NN11	16	1.783
17	○	Ir_CN86_CN86_NN6	17	1.783
18	○	Ir_CN86_CN86_NN2	18	1.783
19	○	Ir_CN74_CN74_NN43	19	1.783
20	○	Ir_CN68_CN68_NN24	20	1.783
21	○	Ir_CN65_CN65_NN47	21	1.783
22	○	Ir_CN65_CN65_NN24	22	1.783
23	○	Ir_CN63_CN63_NN33	23	1.783
24	○	Ir_CN63_CN63_NN23	24	1.783
25	○	Ir_CN63_CN63_NN8	25	1.783

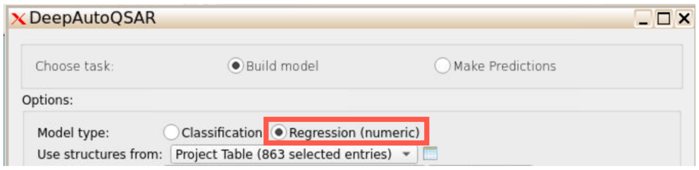
Entries: 863 total, 863 selected, 1 included **Groups:** 1 total, 1 selected **Properties:** 9 total, 2 shown

Figure 2-6. Viewing the rate constants in the Project Table.

Each entry has a rate constant associated with it. These can be visualized in the Project Table ( ). Use the Property Tree ( ) to add the rate constant property (under **All > Materials Science > Secondary > rate constant**)

3. Building a Machine Learning Model Using DeepAutoQSAR

In this section, we will use the DeepAutoQSAR panel to train a machine learning model for rate constant prediction. For a complete description of how AutoQSAR automatically tests various models and makes selections, visit the Machine Learning for Materials Science tutorial.



Choose task: ● Build model ○ Make Predictions

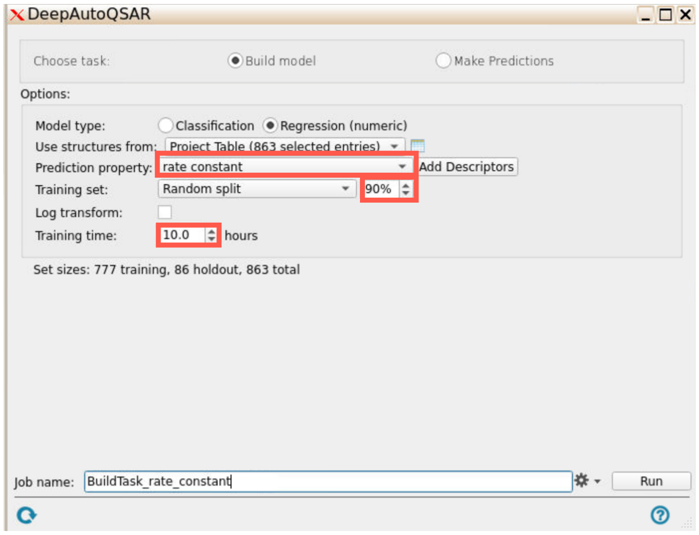
Options:

Model type: ○ Classification ● Regression (numeric)

Use structures from: Project Table (863 selected entries)

Figure 3-1. Choosing task and options.

1. Ensure that all 863 entries are selected from the entry list (use **Shift + Click** or click on the entry group header)
2. Go to **Tasks > Browse All > Discovery Informatics and QSAR > DeepAutoQSAR**
    - The DeepAutoQSAR panel opens
3. Ensure that *Build model* is checked
4. For *Model type*, choose **Regression**
    - Because the data is numerical and continuous, we use the regression *Model type*
5. Ensure that for *Use structures from*, **Project Table (863 selected entries)** is chosen



DeepAutoQSAR

Choose task: ● Build model ○ Make Predictions

Options:

Model type: ○ Classification ● Regression (numeric)

Use structures from: Project Table (863 selected entries)

Prediction property: rate constant

Training set: Random split 90%

Log transform: □

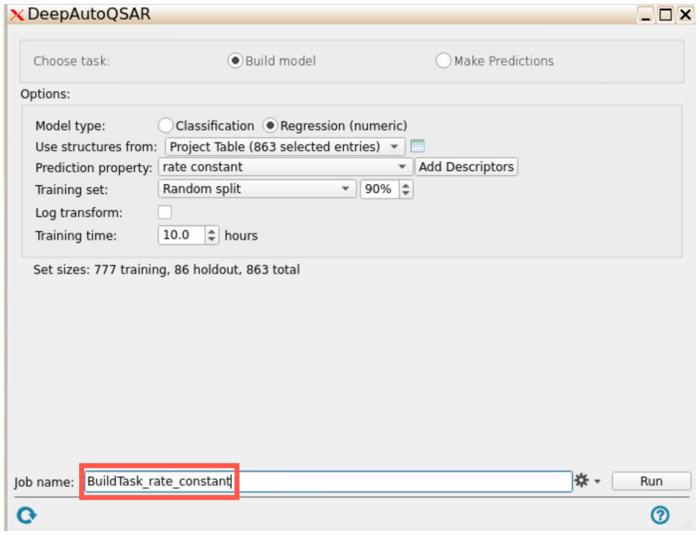
Training time: 10.0 hours

Set sizes: 777 training, 86 holdout, 863 total

Job name: BuildTask_rate_constant

Figure 3-2. Defining the training.

6. Change the *Prediction property* dropdown to **rate constant**
7. Set **90%** for the *Random split*
    - This is the percentage of data to set aside between train and test sets, where 90% of the data is used to train the model and 10% of the data is used to test the model
    - This is a relatively large data set. The 90:10 split ensures that there is significantly more data in the training set than the test set, but still enough data in the test set to assess model performance
8. Set the *Training time* to **10 hours**
    - For datasets with >800 inputs like this one, a 10 hour training time is sufficient to ensure the best models are determined



DeepAutoQSAR

Choose task: ● Build model ○ Make Predictions

Options:

Model type: ○ Classification ● Regression (numeric)

Use structures from: Project Table (863 selected entries)

Prediction property: rate constant

Training set: Random split 90%

Log transform: □

Training time: 10.0 hours

Set sizes: 777 training, 86 holdout, 863 total

Job name: BuildTask_rate_constant

Figure 3-3. Naming the job.

9. Change the *Job name* to **BuildTask_rate_constant**

DO NOT RUN

The job would run for 10 hours as prescribed. The provided data is available for proceeding in **Section 4**. You can proceed to **Section 4** where steps are provided for importing the pre-computed models.

10. **Close** the DeepAutoQSAR panel (or simply move it to the side of your window – we will return to it in a moment)

4. Viewing the Machine Learning Model and Predicting

Using the DeepAutoQSAR panel, we can proceed to view the generated models, and use these to make predictions on an unseen data set.

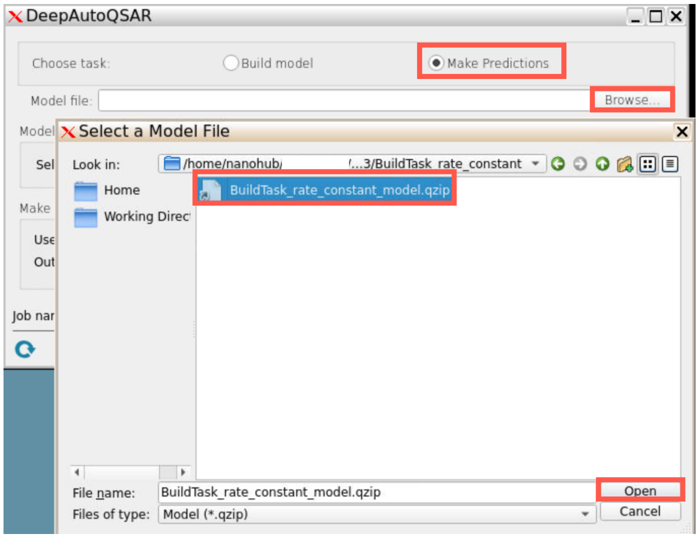
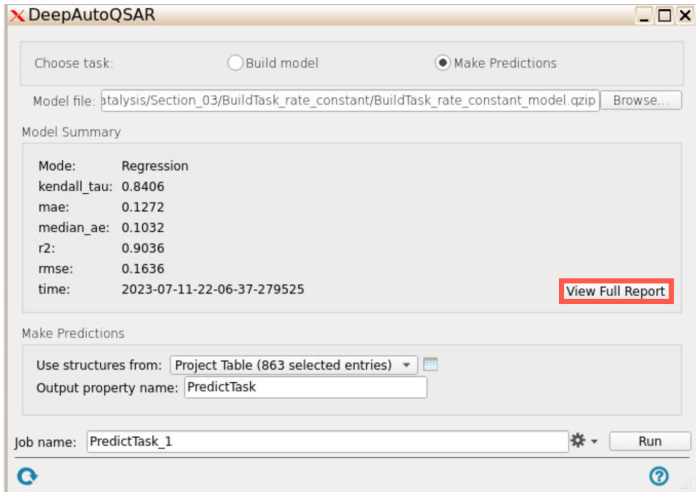

DeepAutoQSAR
Choose task: ○ Build model ● Make Predictions
Model file:
Browse...
Select a Model File
Look in: /home/nanohub/.../3/BuildTask_rate_constant
Home
Working Direc
BuildTask_rate_constant_model.qzip
File name: BuildTask_rate_constant_model.qzip
Files of type: Model (*.qzip)
Open
Cancel

Figure 4-1. Loading the .qzip file.

The output can be analyzed and used for predictions back in the DeepAutoQSAR panel:

1. Return to **Tasks > Browse All > Discovery Informatics and QSAR > DeepAutoQSAR**
   - The [DeepAutoQSAR](#) panel opens
2. For *Choose task*, switch to **Make Predictions**
3. To choose the *Model file* click **Browse**, navigate to the `Section_03 > BuildTask_rate_constant > BuildTask_rate_constant_model.qzip` file and click **Open**
   - The panel will parse the .qzip file and the *Model Summary* section will be populated


DeepAutoQSAR
Choose task: ○ Build model ● Make Predictions
Model file: atalysis/Section_03/BuildTask_rate_constant/BuildTask_rate_constant_model.qzip
Browse...
Model Summary
View Full Report
Make Predictions
Run

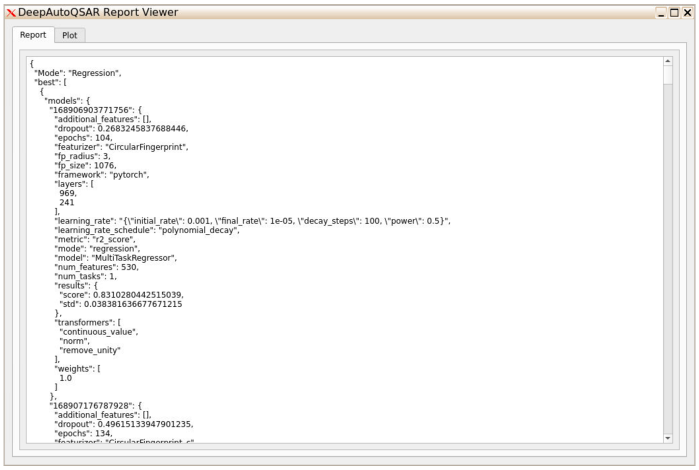
Mode: Regression
kendall_tau: 0.8406
mae: 0.1272
median_ae: 0.1032
r2: 0.9036
rmse: 0.1636
time: 2023-07-11-22-06-37-279525

Use structures from: Project Table (863 selected entries)
Output property name: PredictTask
Job name: PredictTask_1

Figure 4-2. Viewing the Model Summary.

Begin by analyzing the *Model Summary* output. The data presented is a summary of the statistics of the model on the test set. We observe that the DeepAutoQSAR achieves a high R^2 of ~0.90 (denoted as r2) and low root-mean-squared error (rmse) of ~0.16 (an ideal model would have R^2 of 1 and RMSE of 0).

4. Click **View Full Report**



```
{
 "Mode": "Regression",
 "best": [
  {
   "models": {
    "168906903771756": {
     "additional_features": [],
     "dropout": 0.2683245837688446,
     "epochs": 104,
     "featurizer": "CircularFingerprint",
     "fp_radius": 3,
     "fp_size": 1076,
     "framework": "pytorch",
     "layers": [
      969,
      241
     ],
     "learning_rate": "{\"initial_rate\": 0.001, \"final_rate\": 1e-05, \"decay_steps\": 100, \"power\": 0.5}",
     "learning_rate_schedule": "polynomial_decay",
     "metric": "r2_score",
     "mode": "regression",
     "model": "MultiTaskRegressor",
     "num_features": 530,
     "num_tasks": 1,
     "results": {
      "score": 0.8310280442515039,
      "std": 0.038381636677671215
     },
     "transformers": [
      "continuous_value",
      "norm",
      "remove_unity"
     ],
     "weights": [
      1.0
     ]
    },
    "168907176787928": {
     "additional_features": [],
     "dropout": 0.49615133947901235,
     "epochs": 134,
     "featurizer": "CircularFingerprint_c"
```

The *Report* tab includes a raw copy of the JSON output of DeepAutoQSAR. This report contains information on the top four best-performing model ensembles, including their metrics, the classification method used (e.g. dNN, random forest, etc.) and relevant model meta-parameters.

For a complete description of how AutoQSAR automatically tests various models and makes selections, visit the [Machine Learning for](#)

Figure 4-3. Viewing the Report tab.

Materials Science tutorial.

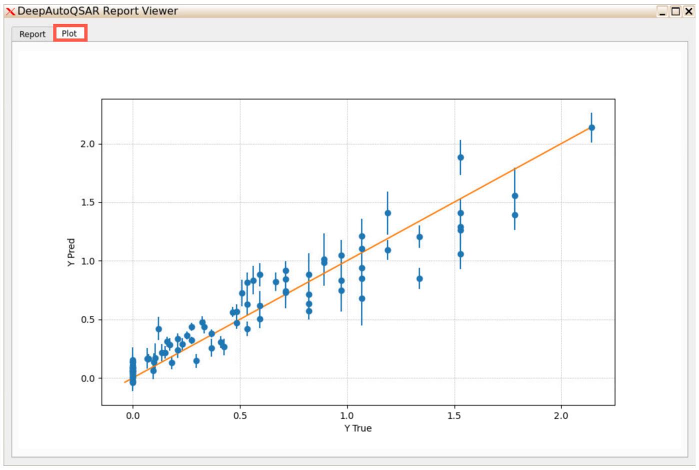
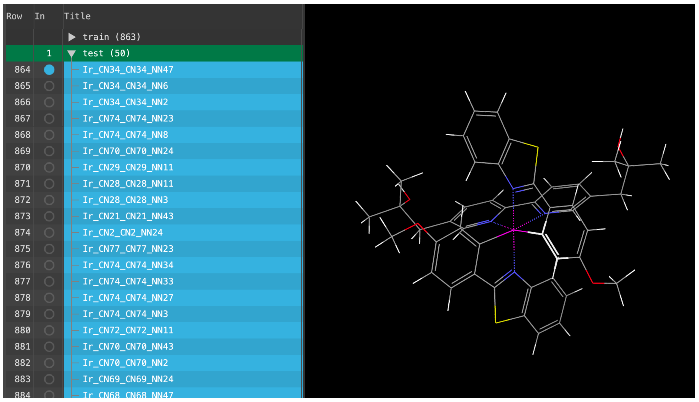


Figure 4-4. Viewing the parity plot.

5. Click on the **Plot** tab

For regression models such as this, the *Plot* tab shows a parity plot.

6. **Close** the DeepAutoQSAR Report Viewer

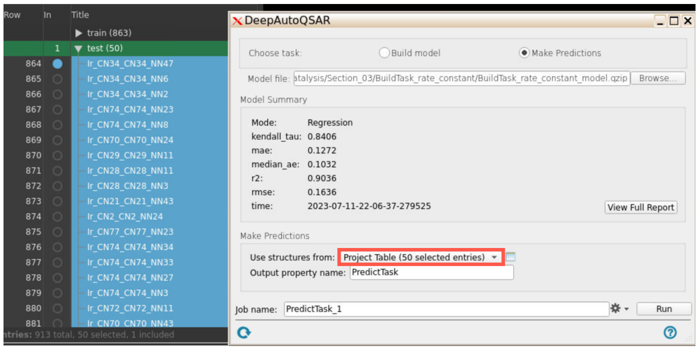


Row	In	Title
		▶ train (863)
	1	▼ test (50)
864	●	Ir_CN34_CN34_NN47
865	○	Ir_CN34_CN34_NN6
866	○	Ir_CN34_CN34_NN2
867	○	Ir_CN74_CN74_NN23
868	○	Ir_CN74_CN74_NN8
869	○	Ir_CN70_CN70_NN24
870	○	Ir_CN29_CN29_NN11
871	○	Ir_CN28_CN28_NN11
872	○	Ir_CN28_CN28_NN3
873	○	Ir_CN21_CN21_NN43
874	○	Ir_CN2_CN2_NN24
875	○	Ir_CN77_CN77_NN23
876	○	Ir_CN74_CN74_NN34
877	○	Ir_CN74_CN74_NN33
878	○	Ir_CN74_CN74_NN27
879	○	Ir_CN74_CN74_NN3
880	○	Ir_CN72_CN72_NN11
881	○	Ir_CN70_CN70_NN43
882	○	Ir_CN70_CN70_NN2
883	○	Ir_CN69_CN69_NN24
884	○	Ir_CN68_CN68_NN47

Figure 4-5. Imported structures in the entry list.

Now, we will use the trained model to make predictions on an unseen data set of iridium complexes that were not in the training data. These complexes have known rate constants from the same experimental study, which we can use to assess the quality of the model for making predictions outside the training set.

7. **Close** the DeepAutoQSAR panel (or simply move it to the side of your window – we will return to it in a moment)
8. Go to **File > Import Structures**
9. Navigate to where you downloaded the provided tutorial files (presumably in your working directory), choose `test.mae` and click **Open**
   - A new entry group is added to the entry list titled test (50)



Row	In	Title
		▶ train (863)
	1	▼ test (50)
864	●	Ir_CN34_CN34_NN47
865	○	Ir_CN34_CN34_NN6
866	○	Ir_CN34_CN34_NN2
867	○	Ir_CN74_CN74_NN23
868	○	Ir_CN74_CN74_NN8
869	○	Ir_CN70_CN70_NN24
870	○	Ir_CN29_CN29_NN11
871	○	Ir_CN28_CN28_NN11
872	○	Ir_CN28_CN28_NN3
873	○	Ir_CN21_CN21_NN43
874	○	Ir_CN2_CN2_NN24
875	○	Ir_CN77_CN77_NN23
876	○	Ir_CN74_CN74_NN34
877	○	Ir_CN74_CN74_NN33
878	○	Ir_CN74_CN74_NN27
879	○	Ir_CN74_CN74_NN3
880	○	Ir_CN72_CN72_NN11
881	○	Ir_CN70_CN70_NN43

ntries: 913 total, 50 selected, 1 included

DeepAutoQSAR

Choose task: ○ Build model ● Make Predictions

Model file: atalysis/Section_03/BuildTask_rate_constant/BuildTask_rate_constant_model.qzip Browse...

Model Summary

Mode: Regression
kendall_tau: 0.8406
mae: 0.1272
median_ae: 0.1032
r2: 0.9036
rmse: 0.1636
time: 2023-07-11-22-06-37-279525

View Full Report

Make Predictions

Use structures from: Project Table (50 selected entries)
Output property name: PredictTask

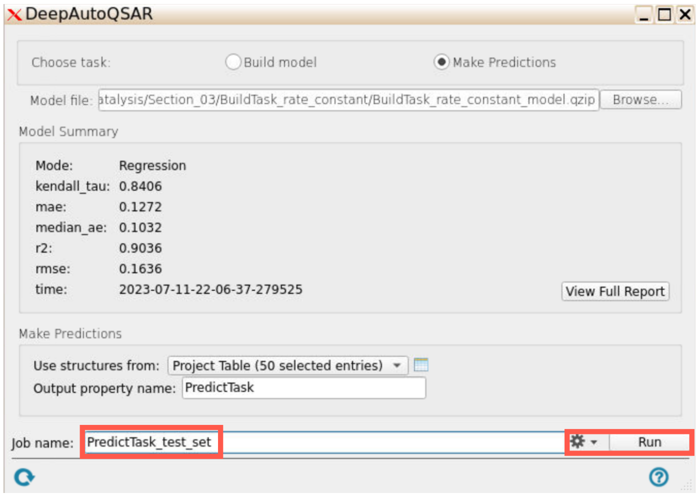
Job name: PredictTask_1 Run

Figure 4-6. The DeepAutoQSAR panel with the

10. Select the entire test (50) group from the entry list
    - Recall that select means to highlight the group in the entry list
11. Return to the DeepAutoQSAR panel
12. Ensure that the panel reflects the progress from the above steps: *Make Predictions* is selected, the .qzip file is loaded and the *Model Summary* is shown

prediction set selected.

13. In the *Make Predictions* section of the panel, ensure that **Project Table (50 selected entries)** is chosen for *Use structures from*



DeepAutoQSAR

Choose task: ○ Build model ● Make Predictions

Model file: atalysis/Section_03/BuildTask_rate_constant/BuildTask_rate_constant_model.qzip Browse...

Model Summary

Mode:	Regression
kendall_tau:	0.8406
mae:	0.1272
median_ae:	0.1032
r2:	0.9036
rmse:	0.1636
time:	2023-07-11-22-06-37-279525

View Full Report

Make Predictions

Use structures from: Project Table (50 selected entries)

Output property name: PredictTask

Job name: PredictTask_test_set

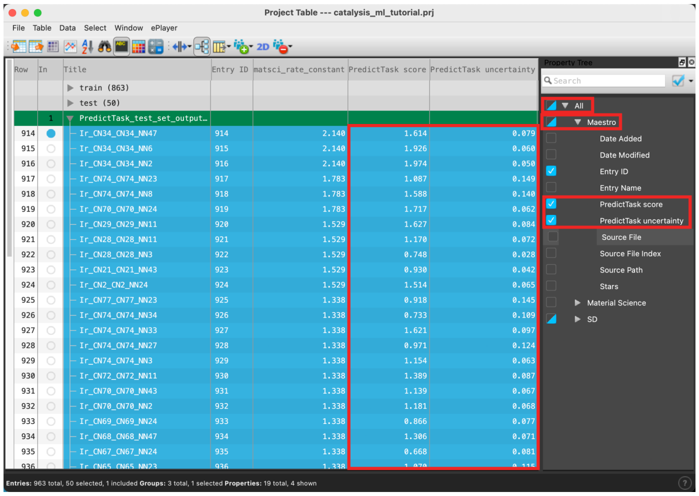
Run

Figure 4-7. Naming and running the job.

14. For *Output property name*, maintain **PredictTask**
    - This will be the name of the predicted property in the project table
15. Change the *Job name* to **PredictTask_test_set**

Adjust the job settings (⚙▾) as needed. This job requires a Linux host. The job can be completed in about 5 minutes, which is of course many orders of magnitude faster than computing the rate constants of 50 systems from first principles. If you do not wish to run the job, feel free to simply import `Section_04 > PredictTask_test_set > PredictTask_test_set_output.maegz` from the provided tutorial files

16. Click **Run**
17. **Close** the DeepAutoQSAR panel



Project Table --- catalysis_ml_tutorial.prj

Row	In	Title	Entry ID	matsci_rate_constant	PredictTask score	PredictTask uncertainty
		▶ train (863)				
		▶ test (50)				
	1	▼ PredictTask_test_set_output…				
914	●	Ir_CN34_CN34_NN47	914	2.140	1.614	0.079
915	○	Ir_CN34_CN34_NN6	915	2.140	1.926	0.060
916	○	Ir_CN34_CN34_NN2	916	2.140	1.974	0.050
917	○	Ir_CN74_CN74_NN23	917	1.783	1.087	0.149
918	○	Ir_CN74_CN74_NN8	918	1.783	1.588	0.140
919	○	Ir_CN70_CN70_NN24	919	1.783	1.717	0.062
920	○	Ir_CN29_CN29_NN11	920	1.529	1.627	0.084
921	○	Ir_CN28_CN28_NN11	921	1.529	1.170	0.072
922	○	Ir_CN28_CN28_NN3	922	1.529	0.748	0.028
923	○	Ir_CN21_CN21_NN43	923	1.529	0.930	0.042
924	○	Ir_CN2_CN2_NN24	924	1.529	1.514	0.065
925	○	Ir_CN77_CN77_NN23	925	1.338	0.918	0.145
926	○	Ir_CN74_CN74_NN34	926	1.338	0.733	0.109
927	○	Ir_CN74_CN74_NN33	927	1.338	1.621	0.097
928	○	Ir_CN74_CN74_NN27	928	1.338	0.971	0.124
929	○	Ir_CN74_CN74_NN3	929	1.338	1.154	0.063
930	○	Ir_CN72_CN72_NN11	930	1.338	1.389	0.087
931	○	Ir_CN70_CN70_NN43	931	1.338	1.139	0.067
932	○	Ir_CN70_CN70_NN2	932	1.338	1.181	0.068
933	○	Ir_CN69_CN69_NN24	933	1.338	0.866	0.077
934	○	Ir_CN68_CN68_NN47	934	1.338	1.306	0.071
935	○	Ir_CN67_CN67_NN24	935	1.338	0.668	0.081
936	○	Ir_CN65_CN65_NN23	936	1.338	1.070	0.115

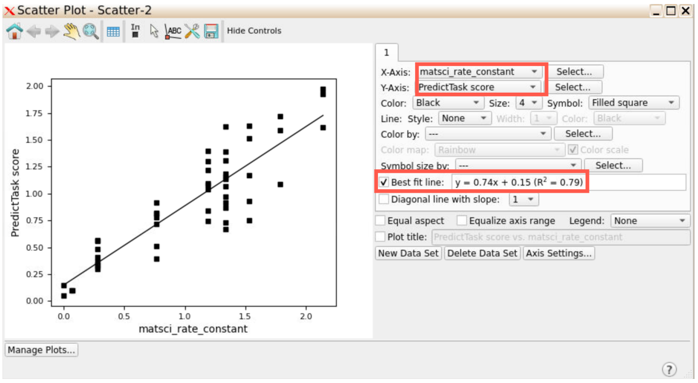
- ☑ ▼ All
  - ☑ ▼ Maestro
    - ☐ Date Added
    - ☐ Date Modified
    - ☑ Entry ID
    - ☐ Entry Name
    - ☑ PredictTask score
    - ☑ PredictTask uncertainty
    - ☐ Source File
    - ☐ Source File Index
    - ☐ Source Path
    - ☐ Stars
  - ☐ ▶ Material Science
  - ☑ ▶ SD

Entries: 963 total, 50 selected, 1 included **Groups:** 3 total, 1 selected **Properties:** 19 total, 4 shown

Figure 4-8. Viewing the output in the Project Table and opening the plots

When the job is complete or after importing, a new entry group is added to the entry list titled PredictTask_test_set_output1 (50) containing the same 50 entries, but now with predicted rate constant property. The data can be analyzed in the Project Table

18. Open the Project Table (TABLE)
19. Use the **Property Tree** ( ) to include the *Predicted Task score* and *uncertainty* properties (**Check** the properties of interest under **All > Maestro > Predict Task score/uncertainty**)

	We can see predicted scores for the various molecules as well as uncertainty values. To compare these values to the known values we will draw a scatter plot. 20. Click the **Manage Plots** ( ) button
 X-Axis: matsci_rate_constant Y-Axis: PredictTask score Color: Black Size: 4 Symbol: Filled square Line: Style: None Width: 1 Color: Black Color by: --- Color map: Rainbow ☑ Color scale Symbol size by: --- ☑ Best fit line: $y = 0.74x + 0.15$ ($R^2 = 0.79$) ☐ Diagonal line with slope: 1 ☐ Equal aspect ☐ Equalize axis range Legend: None ☐ Plot title: PredictTask score vs. matsci_rate_constant Scatter Plot - Scatter-2 Hide Controls 1 Select... New Data Set Delete Data Set Axis Settings... Manage Plots... PredictTask score matsci_rate_constant 0.00 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00 0.0 0.5 1.0 1.5 2.0 *Figure 4-9. A scatter plot of the predicted data versus the known values for the test set.*	21. Click **New Scatter Plot** 22. For *X-Axis* select **matsci_rate_constant** ○ These are the actual values of the target property 23. For *Y-Axis* select **PredictTask score** ○ These are the ML predicted values 24. **Check** *Best fit line* ○ A regression line and equation is added This scatter plot was generated from the values in `PredictTask_test_set_output.maegz`. If you performed this calculation your scatter plot and best fit line will differ slightly.

This workflow highlights the computational efficiency achieved when using ML approaches as compared to other computational (*e.g. ab initio* calculations) or experimental approaches. While this tutorial uses a relatively small dataset, one could expect that a larger training set would further improve prediction accuracy.

5. Conclusion and References

In this tutorial, we learned how to use the DeepAutoQSAR panel to build machine learning models to predict experimentally determined rate constants for a series of iridium complexes. The DeepAutoQSAR model can generalize to unseen data sets and generate fast predictions (~seconds-minutes) as compared to *ab initio* or experimental measurements (~hours-days), enabling the screening of catalysts for enhanced reaction rates. While this tutorial focuses on reaction rate constants of iridium complexes, the workflow can be extended to other catalyst types and properties.

For further learning:

For introductory content, focused on navigating the Schrödinger Materials Science interface, an Introduction to Materials Science Maestro tutorial is available. Please visit the materials science training website for access to 50+ tutorials. For scientific inquiries or technical troubleshooting, submit a ticket to our Technical Support Scientists at help@schrodinger.com.

For self-paced, asynchronous, online courses in Materials Science modeling, including access to Schrödinger software, please visit the Schrödinger Online Learning portal on our website.

For some related practice, proceed to explore other relevant tutorials:

- For more machine learning:
    - Machine Learning for Materials Science
    - Polymer Descriptors for Machine Learning
    - Periodic Descriptors for Inorganic Solids
    - Optoelectronics Active Learning
    - Machine Learning for Sweetness
    - Machine Learning Property Prediction
    - Machine Learning for Ionic Conductivity
- For transition state searching with quantum mechanical methods in molecular or periodic systems:
    - Locating Transition States: Part 1
    - Locating Transition States: Part 2
    - Reaction Workflow with Polyethylene Insertion

For further reading:

- Help documentation on DeepAutoQSAR
- High-throughput Synthesis and Screening of Iridium(III) Photocatalysts for the Fast and Chemoselective Dehalogenation of Aryl Bromides. DOI:10.1021/acscatal.0c02247
- DeepAutoQSAR: Scalable, Intuitive, Deep-learning QSAR models for Big Data Applications (Schrödinger white paper)
- DeepAutoQSAR Hardware Benchmark (Schrödinger white paper)
- Design of Organic Electronic Materials With a Goal-Directed Generative Model Powered by Deep Neural Networks and High-Throughput Molecular Simulations. DOI:10.3389/fchem.2021.800370
- Active Learning Accelerates Design and Optimization of Hole-Transporting Materials for Organic Electronics. DOI:10.3389/fchem.2021.800371
- Some recent publications applying machine learning methods in catalysis and reactivity:
    - Machine Learning in Catalysis, From Proposal to Practicing. DOI:10.1021/acsomega.9b03673
    - Accelerated dinuclear palladium catalyst identification through unsupervised machine learning. DOI:10.1126/science.abj0999

- Univariate classification of phosphine ligation state and reactivity in cross-coupling catalysis. DOI:10.1126/science.abj4213
- Catalytic Performance of Cycloalkyl-Fused Aryliminopyridyl Nickel Complexes towards Ethylene Polymerization by QSPR Modeling. DOI:10.3390/catal11080920

6. Glossary of Terms

Entry List - a simplified view of the Project Table that allows you to perform basic operations such as selection and inclusion

Included - the entry is represented in the Workspace, the circle in the In column is blue

Project Table - displays the contents of a project and is also an interface for performing operations on selected entries, viewing properties, and organizing structures and data

Recent actions - This is a list of your recent actions, which you can use to reopen a panel, displayed below the Browse row. (Right-click to delete.)

Scratch Project - a temporary project in which work is not saved, closing a scratch project removes all current work and begins a new scratch project

Selected - (1) the atoms are chosen in the Workspace. These atoms are referred to as "the selection" or "the atom selection". Workspace operations are performed on the selected atoms. (2) The entry is chosen in the Entry List (and Project Table) and the row for the entry is highlighted. Project operations are performed on all selected entries

Working Directory - the location where files are saved

Workspace - the 3D display area in the center of the main window, where molecular structures are displayed