Machine Learning for Materials Science

Machine Learning for Materials Science

Topics: [Organic Electronics](), [Catalysis and Reactive Systems](), [Informatics and Team Collaboration]()

Methodology: [Machine Learning]()

Products Used: [MS Maestro](), [AutoQSAR]()

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 about AutoQSAR, a tool for automated creation, validation and application of QSPR models following a best practices approach. We will demonstrate the use of AutoQSAR to build and rank order numerical QSPR models, visualize atomic contributions to property predictions and use these models to make predictions on new, unseen datasets.

Tutorial Content
1. Introduction to AutoQSAR
2. Creating Projects and Importing Structures
3. Building a Numerical QSPR Model using AutoQSAR
4. Analyzing a QSPR Model for TADF Singlet-Triplet Energy Splitting
5. Visualizing and Analyzing KPLS Models
6. Using the Model to Make Predictions
7. Conclusions and References
8. Glossary of Terms

1. Introduction to AutoQSAR

Developing Quantitative Structure-Property Relationships (QSPR), also known as Quantitative Structure-Activity Relationships (QSAR), is a powerful technique that is widely used in materials design and drug discovery. QSPR involves finding a mathematical expression that relates the materials' structure to its property for a series of compounds. In practice, developing QSPR models is typically divided into two steps: model training and testing. Model training is when we input the structures and properties from the training set to generate a QSPR model, whereas model testing is when we input a set of new, unseen structures from the testing set to evaluate whether the model could accurately predict properties for novel compounds.

Generating QSPR models for a large range of structures and properties is often time-consuming and usually requires QSPR expertise. To facilitate the development of QSPR models, Schrödinger has automated the creation of accurate QSPR models using a tool called AutoQSAR. AutoQSAR allows the user to modify settings for QSPR modeling, but it does not require the user to have an expert background in QSPR.

AutoQSAR is designed to provide a 'QSPR expertise out-of-the-box' experience by facilitating an automated creation and application of QSPR models using a set of best practices. The best practices include the generation of descriptors, feature selection, creation of a large number of QSPR models with different train/test set splits from multiple machine learning methods, and finally, the performance based ranking of these QSPR models. Predictions can then be made from a particular top ranked QSPR model or from a consensus of the top scoring models. The AutoQSAR workflow is summarized below:
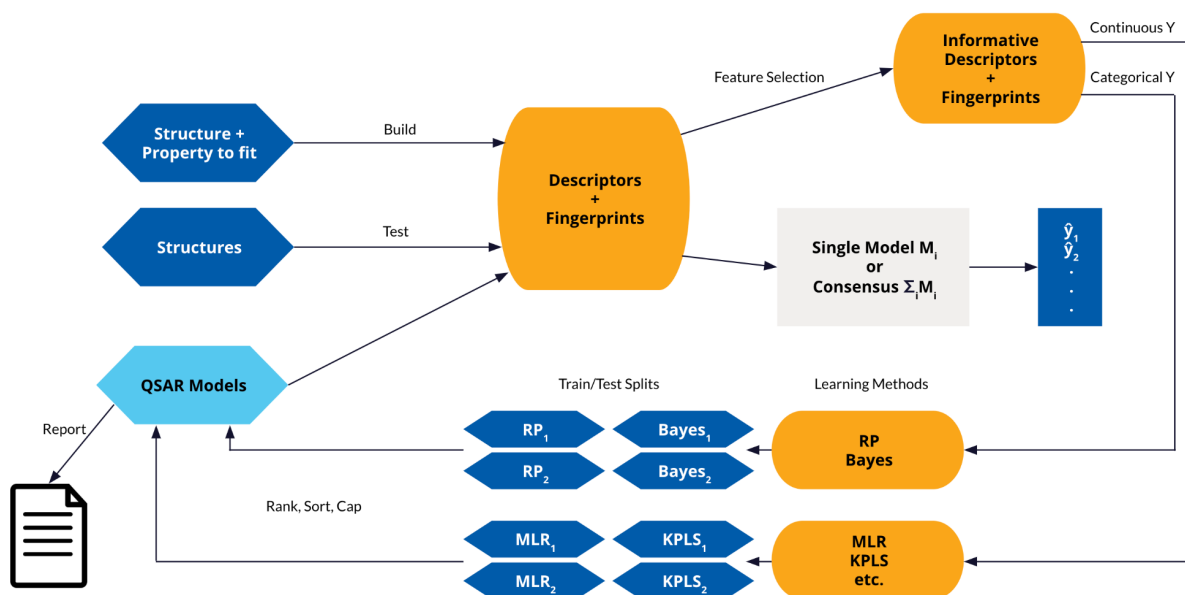

Figure 1. Workflow of AutoQSAR to develop QSPR models. For model training, structures and properties are inputted into AutoQSAR. Descriptors and fingerprints are then computed based on the structure and subsequently used to train machine learning models to predict either continuous or categorical properties. A series of train/test splits and machine learning models are then used to

identify the best QSPR model for the property of interest. For model testing, descriptors and fingerprints are computed for the unseen structures, which are then inputted into a pre-trained AutoQSAR model to generate property predictions.

AutoQSAR Key Features

- AutoQSAR takes in 1D, 2D or 3D structures as input and any desired property to create either numerical or categorical models.
- It uses topology-based descriptors including estate counts ([electrotopological state indices](#)), 2D topological descriptors, functional group counts and 4 types of fingerprints (dendritic, linear, radial and MOLPRINT2D). In addition, you can also add your own descriptors.
- For feature selection, AutoQSAR eliminates descriptors where >90% of the training set has the same value. AutoQSAR also ensures that no pair of descriptors are linearly correlated by eliminating descriptors with an absolute Pearson's *r* correlation coefficient greater than 0.8 to another descriptor. Additionally, for the fingerprints, only the most significant 10,000 bits with the greatest variance over the training set are employed.
- For numerical models, the machine learning methods employed are Partial Least Squares regression (PLS), best subset Multiple Linear Regression (MLR), kernel-based PLS (kPLS) and Principal Components Regression (PCR).
- For categorical models, AutoQSAR uses Naive Bayes classification and ensemble recursive partitioning.
- AutoQSAR rank orders all the QSPR models by their predictive accuracy and retains only the top 10 models by default. The total number of top models retained is editable by the user.
- When predicting properties of new structures, AutoQSAR estimates whether the new structure falls within the applicability domain by comparing the structural similarity between the new structure and the original training set. AutoQSAR will output a domain score to indicate whether the new structure lies within or outside the applicability domain. Domain alert of one indicates the new structure is outside the applicability domain of the model.
- AutoQSAR does not create neural network models as they are found to overfit too easily for small datasets.

Data Curation Recommendations for AutoQSAR:

A few considerations are summarized below with respect to the input structures used in AutoQSAR:

- There should be a consistency in the input structures, such as consistent representations for dative or other special type(s) of bonds for organometallic species. Also, ensure removal of duplicates and salts.
- Data heterogeneity, such as data from different species/protocols, should be avoided because that may lead to poor models.
- Data inadequacy should be minimized - the predicted property or descriptors should not span multiple orders of magnitude. If it does, you can convert the property/descriptor values to a logarithmic scale prior to QSPR modeling, e.g. frequencies that span over KHz, MHz and GHz ranges should be converted to logarithmic scale first.

- Ensure the dataset is a reasonable size. An absolute minimum of 20 compounds is recommended. For a very large dataset (i.e. on the order of >5000 compounds), [DeepAutoQSAR](#) is recommended.

In this tutorial, we will apply the AutoQSAR approach to an example in [organic electronic](#) discovery, specifically the discovery of thermally activated delayed fluorescence (TADFs) molecules. TADFs are a 3rd generation class of organic light-emitting display (OLED) materials used in everyday displays, such as smartphones, smartwatches and TVs. These molecules share a common property: a small energy gap between their singlet and triplet energy states, abbreviated ΔE_{ST}. Tuning ΔE_{ST} allows for higher efficiency light emission in the form of delayed fluorescence. Design of new TADFs can lead to breakthroughs in new emitters that do not require heavy metals and have long lifetimes. This tutorial focuses on building QSPR models to predict the singlet-triplet splitting energy (ΔE_{ST}) for a set of TADF molecules using the AutoQSAR panel. After training the model, we will then use it to predict ΔE_{ST} for 14 unseen TADFs. While this tutorial example focuses on an OLED example, the workflow is flexible enough to be applied to different material classes. The workflow of this tutorial is summarized below:
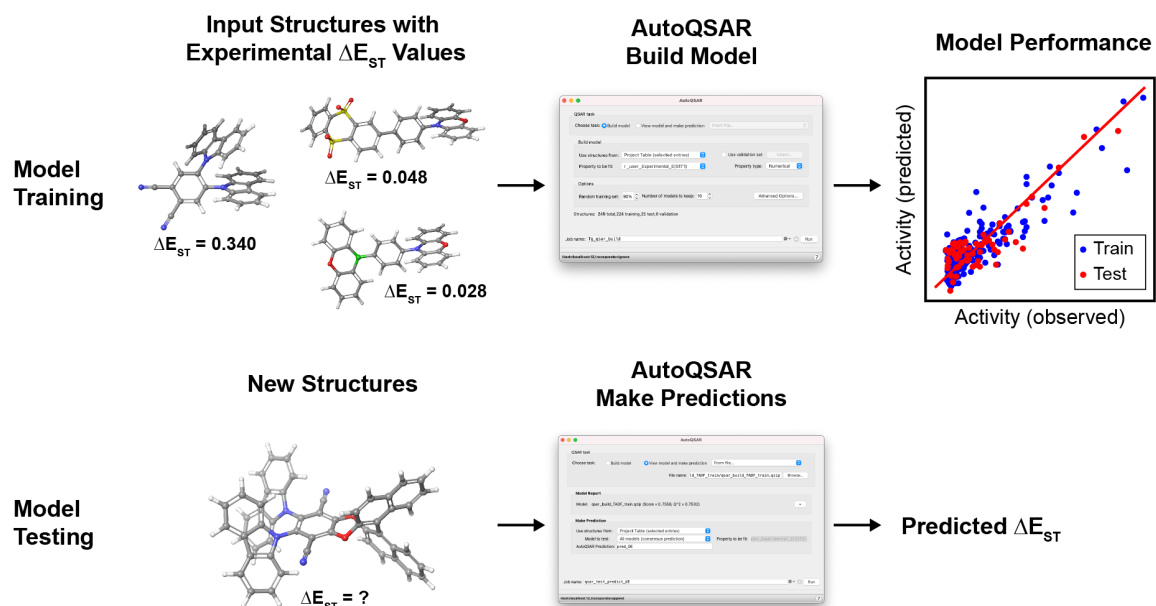

Figure 2. Tutorial workflow showing a subset of the input TADF structures, AutoQSAR panel, and the output predicted versus observed ΔE_{ST}. After training AutoQSAR models, new structures are inputted to generate ΔE_{ST} predictions of novel compounds.

It is recommended to work through this tutorial first to understand the standard AutoQSAR workflow before trying more specialized tutorials, such as [Polymer Descriptors for Machine Learning](#) and [Periodic Descriptors for Inorganic Solids](#). Alternative machine learning approaches like active learning and DeepAutoQSAR are also available: [Optoelectronics Active Learning](#), [Cheminformatics Machine Learning for Homogeneous Catalysis](#) and [Machine Learning for Sweetness](#). To learn about using pre-built machine learning models to predict properties, please refer to the [Machine Learning Property Prediction](#) tutorial.

2. Creating Projects and Importing Structures

At the start of the session, change the file path to your chosen <u>Working Directory</u> in MS Maestro to make file navigation easier. Each session in MS Maestro begins with a default <u>Scratch Project</u>, 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 <u>Entry List</u> and <u>Project Table</u>. The <u>Entry List</u> is located to the left of the <u>Workspace</u>. The <u>Project Table</u> 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.
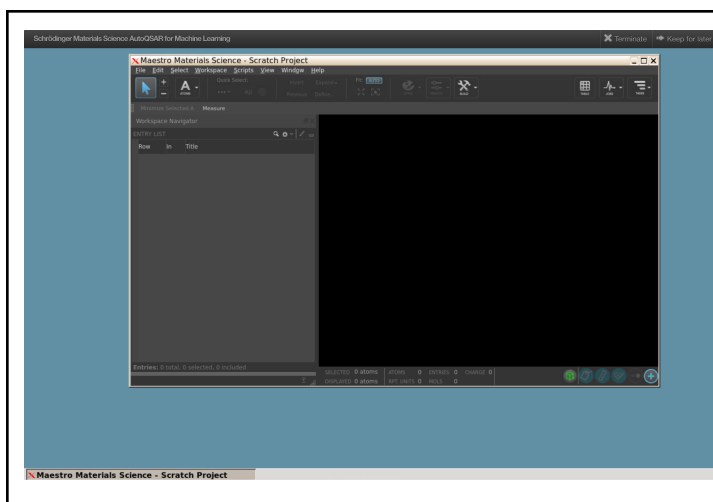
	1. Launch the tool in the nanoHUB interface ○ Launching the tool will automatically open up MS Maestro
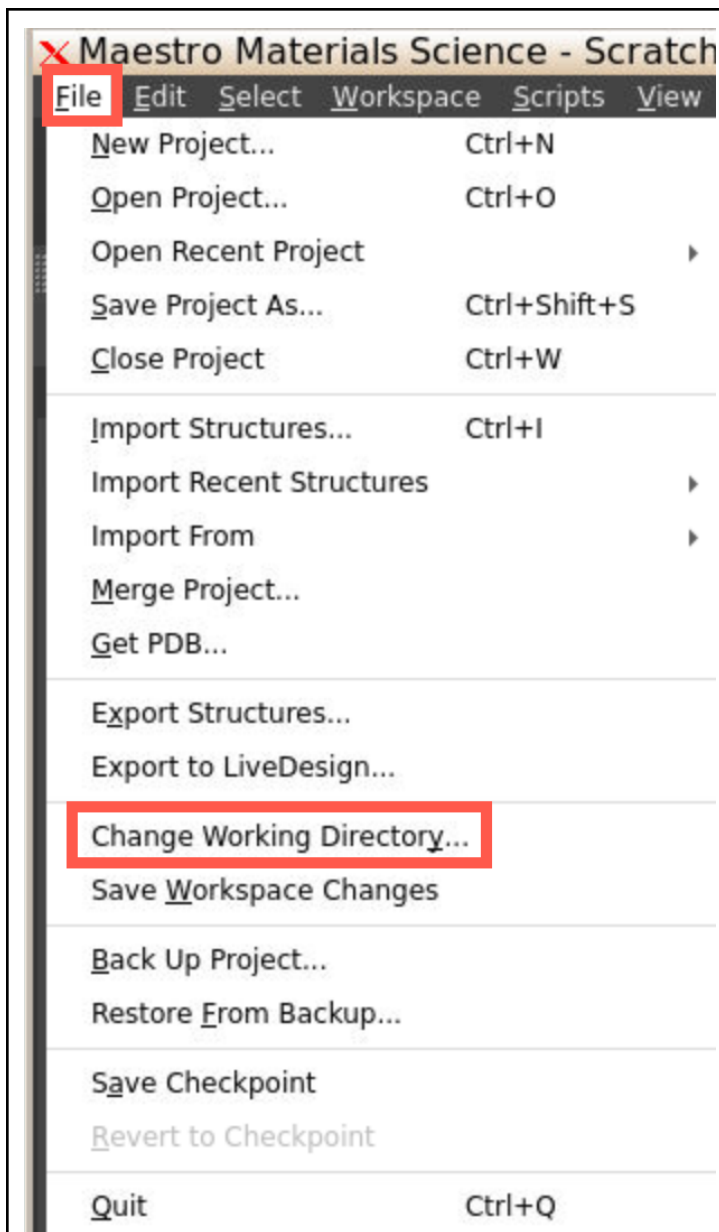
Figure 2-1. Change Working Directory option.
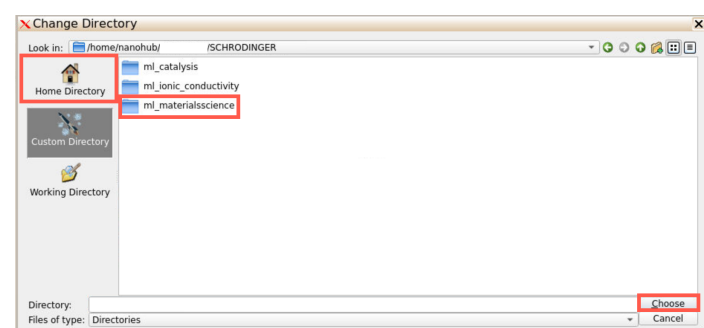
2. Go to **File > Change Working Directory**

Figure 2-2. Selecting the Working Directory.

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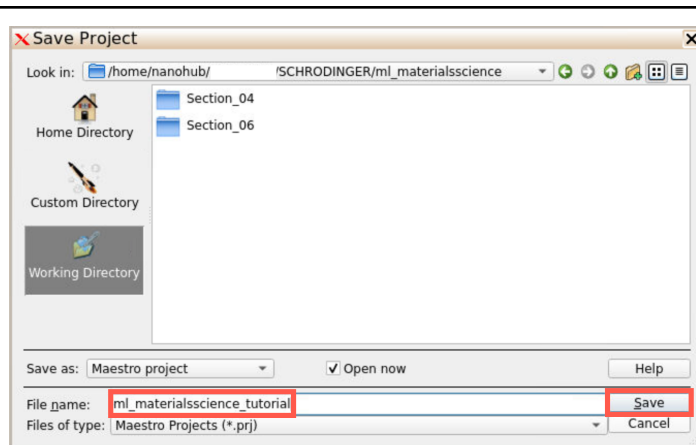   - All files needed to execute this tutorial are included in this directory

Figure 2-3. Save Project panel.

5. Go to **File > Save Project As**
6. Change the *File name* to **ml_materialsscience_tutorial**, click **Save**
   ○ The project is now named `ml_materialsscience_tutorial.prj`
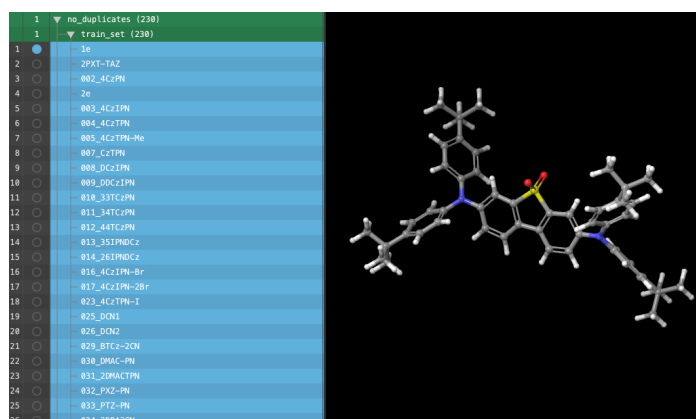
Figure 2-4. The entry list after importing.

We will import a library of 230 TADF molecules:

7. Go to **File > Import Structures**
8. Navigate to where you downloaded the provided tutorial files, **choose** `TADF_train_set.mae` and click **Open**
   ○ A new entry group is added to the <u>entry list</u> containing 230 entries

The 230 TADFs comprising the dataset have experimental ΔE_{ST} values in the range of 0.0-1.1. If you are interested, you can view these values in the <u>Project Table</u> ( ).

3. Building a Numerical QSPR Model using AutoQSAR

Here, we will use the series of TADF molecules with known ΔE_{ST} values as input structures and build a numerical QSPR model with the AutoQSAR panel.
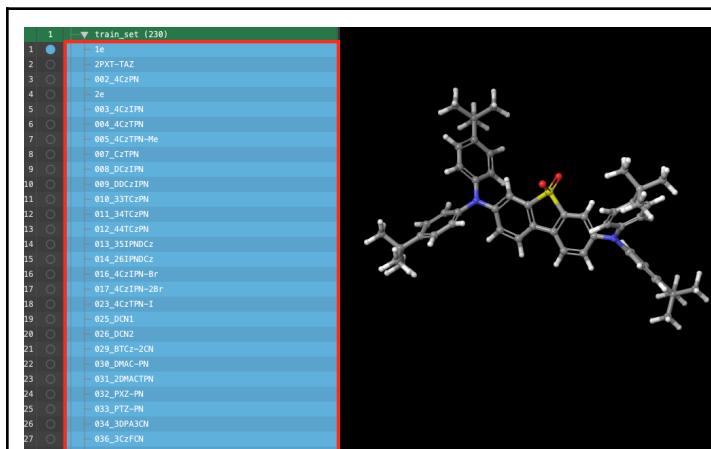

Figure 3-1. Selecting the entire entry group.

1. Select the entire **train_set (230)** entry group by clicking on the group header
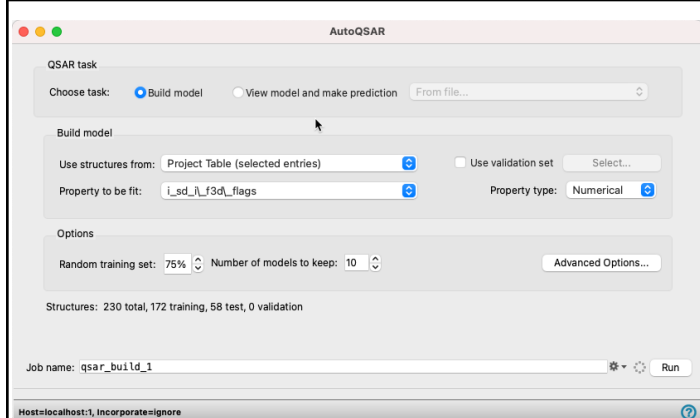   - Recall that selecting means to highlight the entries in the entry list

Figure 3-2. The AutoQSAR panel.

2. Go to **Tasks > Materials > Informatics > AutoQSAR**
   - The AutoQSAR panel opens

Note: This panel is a single entry point for building a model, viewing the model and making predictions
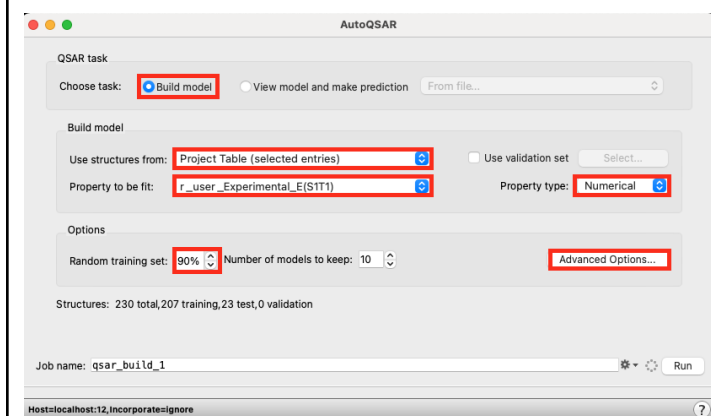

Figure 3-3. Building a model in the AutoQSAR panel.

3. For *Choose task*, ensure that **Build model** is selected
4. In the *Build model* section of the panel, next to *Use structures from*, choose **Project Table (selected entries)**
5. Set the *Property to be fit* to **r_user_Experimental_E(S1T1)**
   - These are the experimental ΔE_{ST} values
6. Maintain *Property type* as **Numerical**
7. Input **90%** for the *Random training set*
   - 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

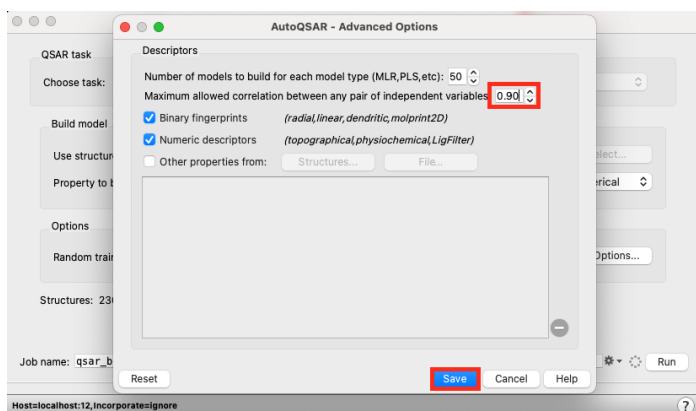|	
	the data is used to test the model ○ With this relatively small data set, the 90:10 split ensures that there is significantly more data in the training set than the test set, but still enough data in the test set to assess model performance 8. Click **Advanced Options**

Figure 3-4. AutoQSAR - Advanced Options.
	9. Maintain **50** for the *Number of models to build for each model type*
	10. Change the *Maximum allowed correlation between any pair of individual variables* to **0.90**
	
	    ○ A higher correlation threshold allows AutoQSAR to use descriptors that are linearly correlated with each other, which may obtain better results
	
	Note: Using the Advanced Options panel, we can select and modify the types of descriptors that will be used to build the model. The default descriptors and fingerprints are for molecular systems only. If interested in using your own descriptors, you can:
	
	    ● Check **Other properties from…** Click **Structures** to choose descriptors from the Project Table. You can uncheck **Binary fingerprints** and **Numeric descriptors** if only custom numeric descriptors should be used. Or,
	    ● Read a list of the properties from a plain text file by locating the file location from **File…**. Each property name must be written on each line, and the name must be exactly as it appears in the structure source
	
	10. Click **Save** to close the Advanced Options panel
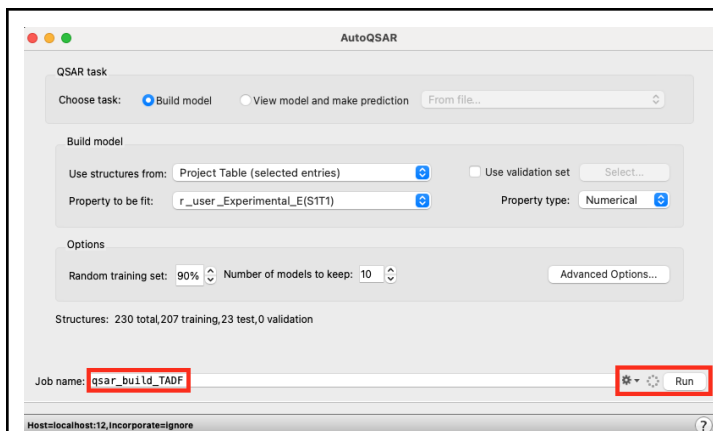
Figure 3-5. The Job Settings panel.

11. Change the *Job name* to **qsar_build_TADF**

12. Adjust the job settings (⚙▾) as needed
    - This job requires a CPU host. The job can be completed in about 20 minutes on a 12 CPU host

13. If you would like to run the job, click **Run**. Otherwise, provided files and instructions to use them are available in **Section 4**

14. **Close** the AutoQSAR panel

Note: AutoQSAR models can also be built from the command line. Visit the AutoQSAR utility to view all the available options.

4. Analyzing a QSPR Model for TADF Singlet-Triplet Energy Splitting

In this section, we analyze the models that were built in the previous section for the ΔE_{ST} of the 230 TADF molecules again using the AutoQSAR panel.
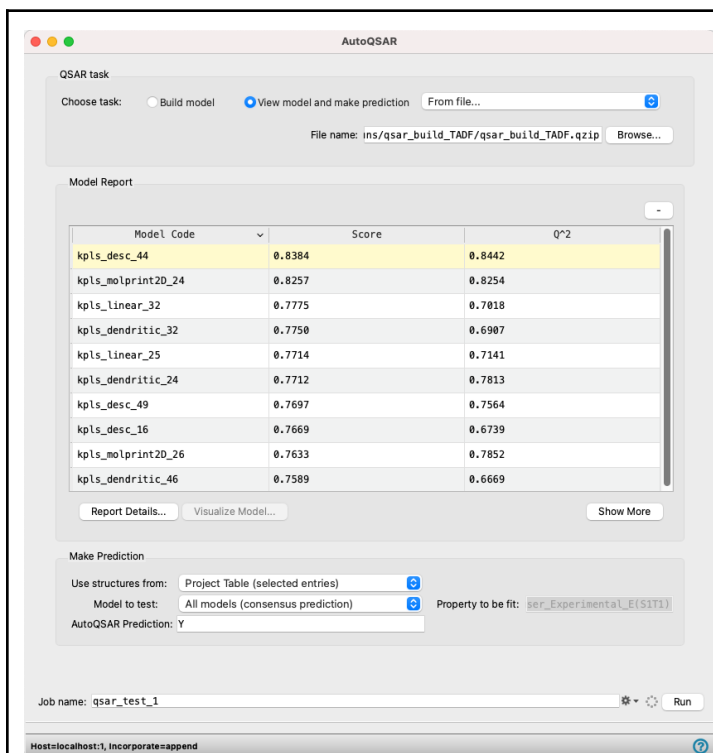

Figure 4-1. Viewing the output models.

If you ran the job, when the job is complete, a banner will appear indicating "Your job qsar_build_TADF has completed" but no new entries will be imported into the entry list. Whether or not you ran the job, you can proceed:
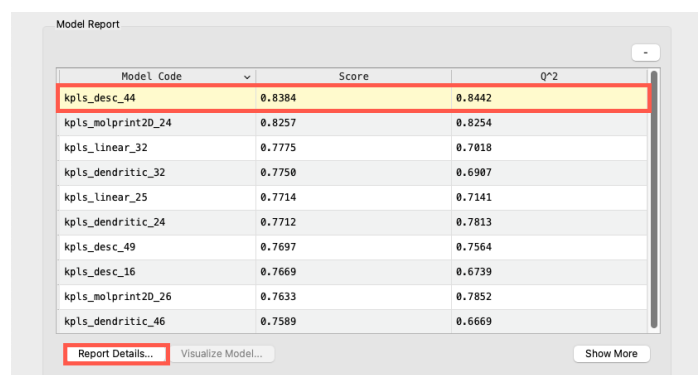
1. Return to **Tasks > Materials > Informatics > AutoQSAR**
   - The AutoQSAR panel opens
2. For *Choose task*, switch to **View model and make prediction**
3. For *File name*, click on **Browse**
   - A panel to select the model set file opens
4. Select the `qsar_build_TADF.qzip` file either from the provided files (`Section_04 > qsar_build_TADF >`

<table>
<tr><td></td><td>

`qsar_build_TADF.qzip`) or from your job directory and click **Open**
- ○ The models are imported into the panel
5. In the *Model Report* section, **click the + button**
  - ○ The *Model Report* section of the panel shows the ranking score and Q^2 value (the R^2 for the test set) for the best models

Note: Use the **Show More/Show Less** button to view additional data columns in the *Model Report* section of the panel

For more detail about how the parameters are calculated, please visit the [help documentation](#).

</td></tr>
<tr><td>

Figure 4-2. Choosing the top model and viewing the Report Details.

</td><td>

6. **Click to Highlight** the best model (the first row by default), which has *Model Code* **kpls_desc_44**
  - ○ The naming indicates that this is a QSPR model that was generated by KPLS fitting with 2D descriptors using the 44th random split of the learning set
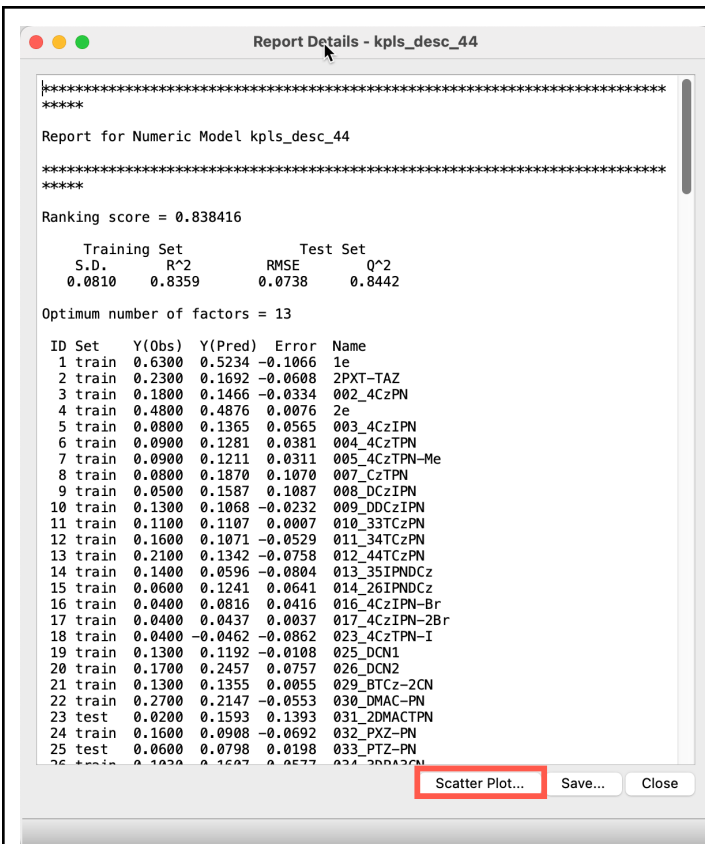7. Click the **Report Details** button

</td></tr>
</table>

```
************************************************************
*****
Report for Numeric Model kpls_desc_44

************************************************************
*****

Ranking score = 0.838416

     Training Set              Test Set
   S.D.       R^2         RMSE        Q^2
  0.0810    0.8359       0.0738      0.8442

Optimum number of factors = 13

 ID Set    Y(Obs)  Y(Pred)  Error   Name
  1 train  0.6300  0.5234 -0.1066  1e
  2 train  0.2300  0.1692 -0.0608  2PXT-TAZ
  3 train  0.1800  0.1466 -0.0334  002_4CzPN
  4 train  0.4800  0.4876  0.0076  2e
  5 train  0.0800  0.1365  0.0565  003_4CzIPN
  6 train  0.0900  0.1281  0.0381  004_4CzTPN
  7 train  0.0900  0.1211  0.0311  005_4CzTPN-Me
  8 train  0.0800  0.1870  0.1070  007_CzTPN
  9 train  0.0500  0.1587  0.1087  008_DCzIPN
 10 train  0.1300  0.1068 -0.0232  009_DDCzIPN
 11 train  0.1100  0.1107  0.0007  010_33TCzPN
 12 train  0.1600  0.1071 -0.0529  011_34TCzPN
 13 train  0.2100  0.1342 -0.0758  012_44TCzPN
 14 train  0.1400  0.0596 -0.0804  013_35IPNDCz
 15 train  0.0600  0.1241  0.0641  014_26IPNDCz
 16 train  0.0400  0.0816  0.0416  016_4CzIPN-Br
 17 train  0.0400  0.0437  0.0037  017_4CzIPN-2Br
 18 train  0.0400 -0.0462 -0.0862  023_4CzTPN-I
 19 train  0.1300  0.1192 -0.0108  025_DCN1
 20 train  0.1700  0.2457  0.0757  026_DCN2
 21 train  0.1300  0.1355  0.0055  029_BTCz-2CN
 22 train  0.2700  0.2147 -0.0553  030_DMAC-PN
 23 test   0.0200  0.1593  0.1393  031_2DMACTPN
 24 train  0.1600  0.0908 -0.0692  032_PXZ-PN
 25 test   0.0600  0.0798  0.0198  033_PTZ-PN
 26 train  0.1030  0.1607  0.0577  034_3DPA2CN
```

Scatter Plot...   Save...   Close

Figure 4-3. Report Details.

A panel opens with a report containing details of the selected KPLS model with the respective experimental and predicted ΔE_{ST} values.
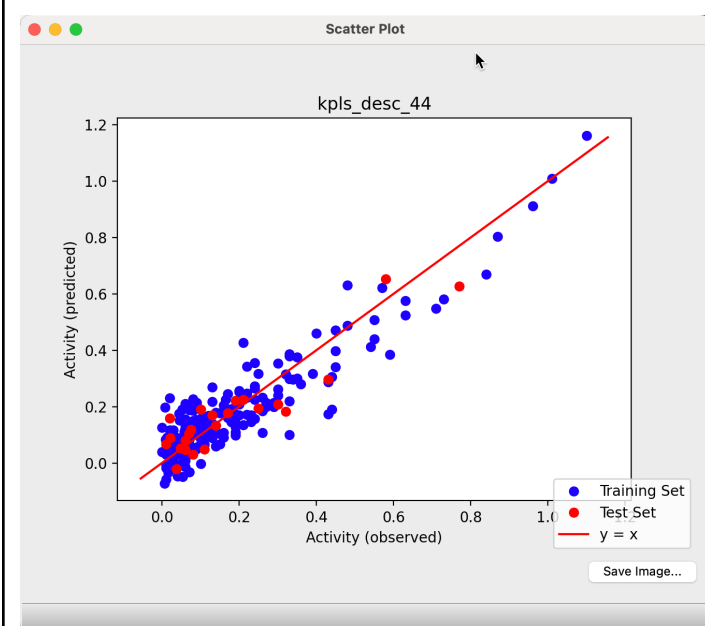
8. Click the **Scatter Plot** button

Figure 4-4. Parity plot showing predicted versus observed ΔE_{ST} for the KPLS QSPR model.
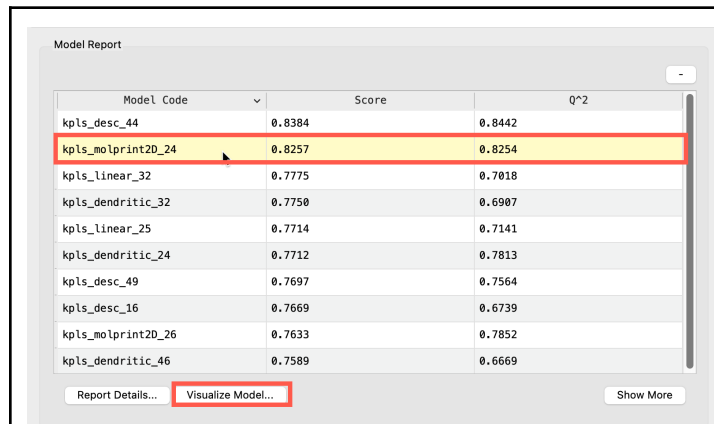
A parity plot of the KPLS QSPR model performance is displayed. An ideal model would have train (blue dots) and test (red dots) set points lie along the red y=x line, indicating that the predictions match the observables.

9. **Close** the Scatter Plot panel
10. **Close** the Report Details panel

5. Visualizing and Analyzing KPLS Models

Interpretability of structure/activity for molecules is highly desirable for any QSPR model. Here we will visualize atomic level contributions to the machine learning model.

Figure 5-1. Visualizing the model.

1. **Return** to the AutoQSAR panel and **Click to Highlight** the **kpls_molprint2D_24** *Model Code*
   - Fingerprint-based models, such as MOLPRINT2D, allow for visualization of atomic contributions
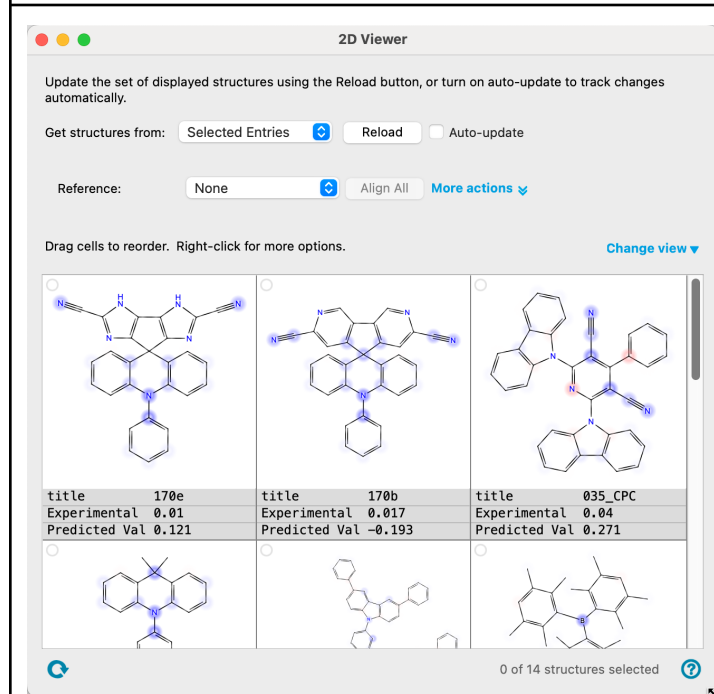2. Click **Visualize Model**
   - The 2D Viewer panel opens

Figure 5-2. 2D Viewer with contribution coloring.

For each structure, each atom that contributed to a fingerprint used in building the model is marked with a colored disk that represents the value of the contribution to the property due to that atom. The disks are blue (reduced ΔE_{ST}) for negative values and red (increased ΔE_{ST}) for positive values. The color saturation indicates the magnitude of the contribution. Atoms that did not appear in any fingerprint are not marked with a disk.
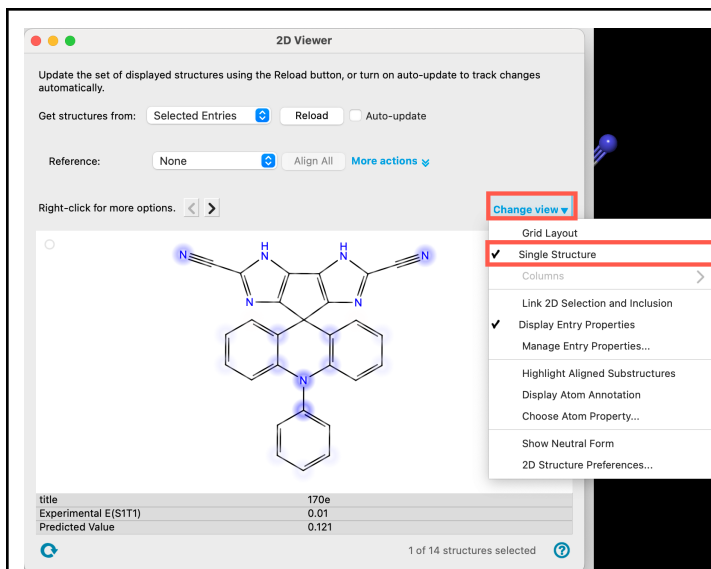
Figure 5-3. Visualizing individual molecules.

We can visualize the atomic contributions for each molecule in further detail

3. In the **Change view** dropdown, choose **Single Structure**
   ○ Molecules are displayed individually in the 2D viewer
4. Navigate between entries by clicking the **right** and **left** arrows

In addition to the coloring, the experimental and predicted values are shown below each frame.

Note: Use the *More actions* dropdown for other navigational options and to generate a report

Note: Select *Link 2D Selection and Inclusion* from the *Change view* dropdown to <u>select</u> the viewed molecule in the <u>entry list</u> and <u>include</u> it in the <u>workspace</u> respectively to interact with MS Maestro from the 2D Viewer panel.

5. **Close** the 2D Viewer

6. Using the Model to Make Predictions

In this final section, we will predict ΔE_{ST} for 14 TADFs whose ΔE_{ST} values are unseen by our QSPR model, but for which we know the experimental ΔE_{ST} values for comparison.
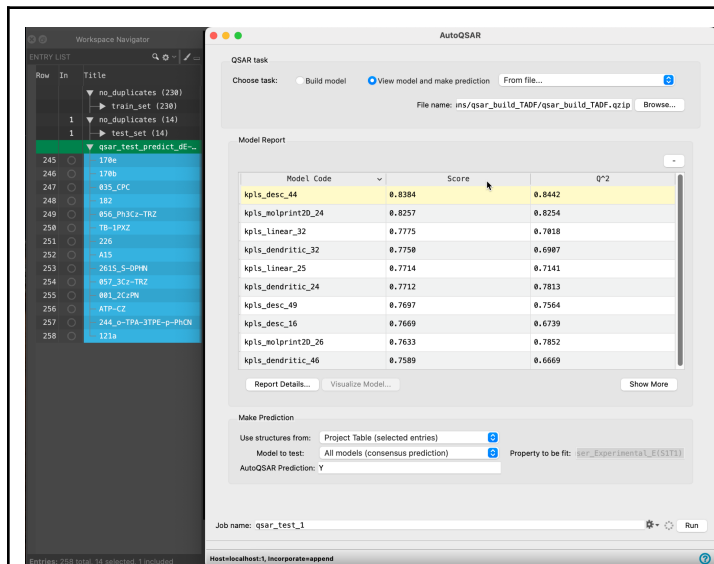
Figure 6-1. Selecting the test data set.

1. Go to **File > Import Structures**
2. Navigate to where you downloaded the provided tutorial files, **choose** `Section_06 > TADF_test_set.mae` and click **Open**
   - A new entry group is added to the <u>entry list</u> containing 14 entries

The 14 TADFs comprising the test set have experimental ΔE_{ST} values also in the range of 0.0-1.1. If you are interested, you can view these values in the <u>Project Table</u> ( )

3. <u>Select</u> the entire **test_set (14)** group in the <u>entry list</u> and return to the AutoQSAR panel
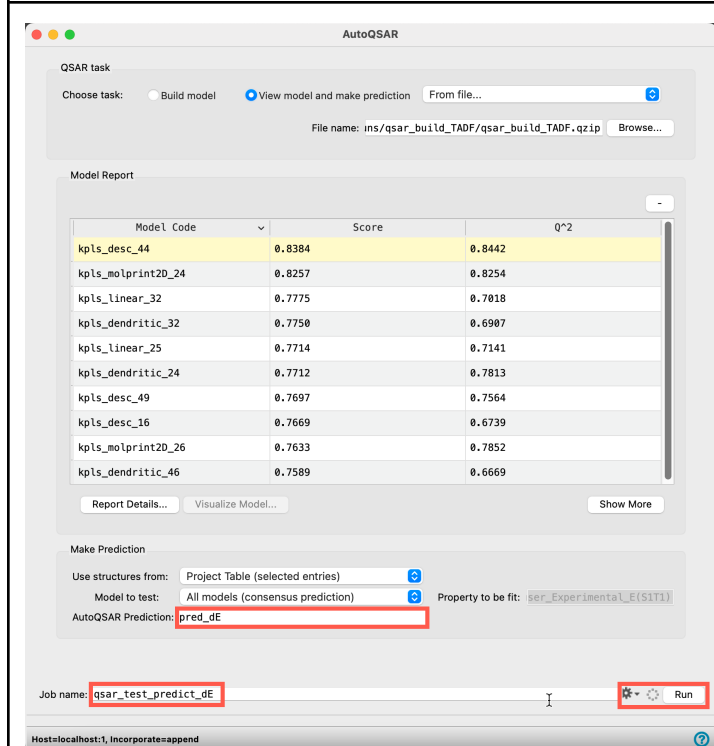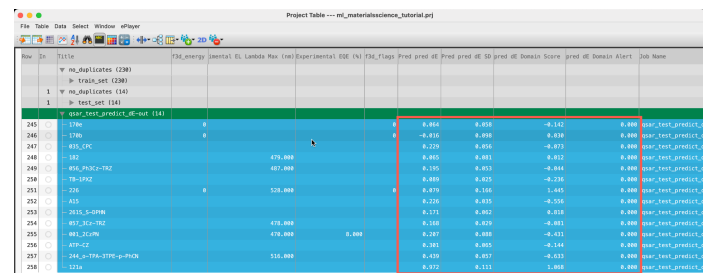
Figure 6-2. Making predictions.

4. Ensure that *Choose task* is still set to **View model and make prediction**
5. Ensure that the *File name* still points to **qsar_build_TADF.qzip**
6. In the *Make Prediction* section, ensure that *Use structures from* is set to **Project Table (selected entries)**
7. Ensure that *Model to test* is set to **All models (consensus prediction)**
   - Consensus prediction averages the results of the retained models, which can often increase the accuracy of the predictions
   - It does not matter which row is highlighted when performing consensus predictions
8. For *AutoQSAR Prediction*, input **pred_dE**
9. Change the *Job name* to **qsar_test_predict_dE**

|	
	10. Adjust the job settings (⚙ ▾) as needed ○ This job requires a CPU host. The job can be completed in about 2 minutes on a 12 CPU host 11. Click **Run**
 Figure 6-3. The predicted values in the Project Table.	When the job is complete, a new entry group is added to the entry list entitled **qsar_test_predict_dE-out1 (14)** containing the same fourteen structures. These structures now have the predicted ΔE_{ST} values associated with them. 12. **Close** the AutoQSAR panel 13. Select the **qsar_test_predict_dE-out1 (14)** entry group from the entry list 14. Open the Project Table (⊞ TABLE) ○ You can view predicted ΔE_{ST} values along with the standard deviations, domain score and domain alert values at the end of the table To compare these values to the known values we will draw a scatter plot. 15. Click the **Manage Plots** (📈) button
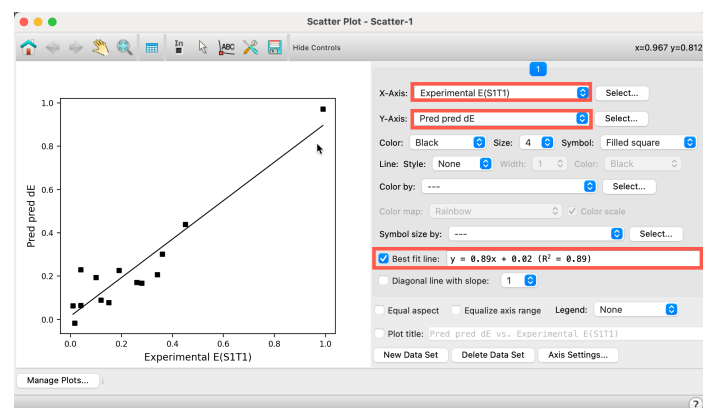
Figure 6-4. A scatter plot of the predicted data versus the known values for the test set.
	16. Click **New Scatter Plot**
	17. For *X-Axis* select **Experimental E(S1T1)**
	    ○ These are the experimental values of the property
	18. For *Y-Axis* select **Pred pred dE**
	    ○ These are the ML predicted values
	19. **Check** *Best fit line*
	    ○ A regression line and equation is added
	
	Feel free to stylize the graph as you wish, and to save an image if you wish.

The best fit line between predicted and actual values shows a reasonable R^2 of 0.89 (an ideal	

model would have an R^2 of 1.00). The results suggest that the ML model derived from the AutoQSAR panel could generalize to unseen TADF molecules. Furthermore, 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 envision a larger training set would further improve prediction accuracy.

7. Conclusion and References

In this tutorial, we learned how to use AutoQSAR to generate accurate QSPR models for the property predictions of OLED materials. We further showed how AutoQSAR can be used to visualize atomic contributions to a particular property. Finally, we showed how AutoQSAR could be used to predict properties for new, unseen structures. Altogether, AutoQSAR provides an automated way of generating accurate machine learning models without in-depth expertise, which could be broadly applied for distinct classes of materials.

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:

- [Polymer Descriptors for Machine Learning](#)
- [Periodic Descriptors for Inorganic Solids](#)
- [Machine Learning for Ionic Conductivity](#)
- [Optoelectronics Active Learning](#)
- [Cheminformatics Machine Learning for Homogeneous Catalysis](#)
- [Machine Learning for Sweetness](#)
- [Machine Learning Property Prediction](#)

For further reading:

- [AutoQSAR](#) help documentation

- 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
- Accelerated design and optimization of OLED materials via active learning. DOI:10.1117/12.2598140
- DeepAutoQSAR Hardware Benchmark (Schrödinger white paper)

8. 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