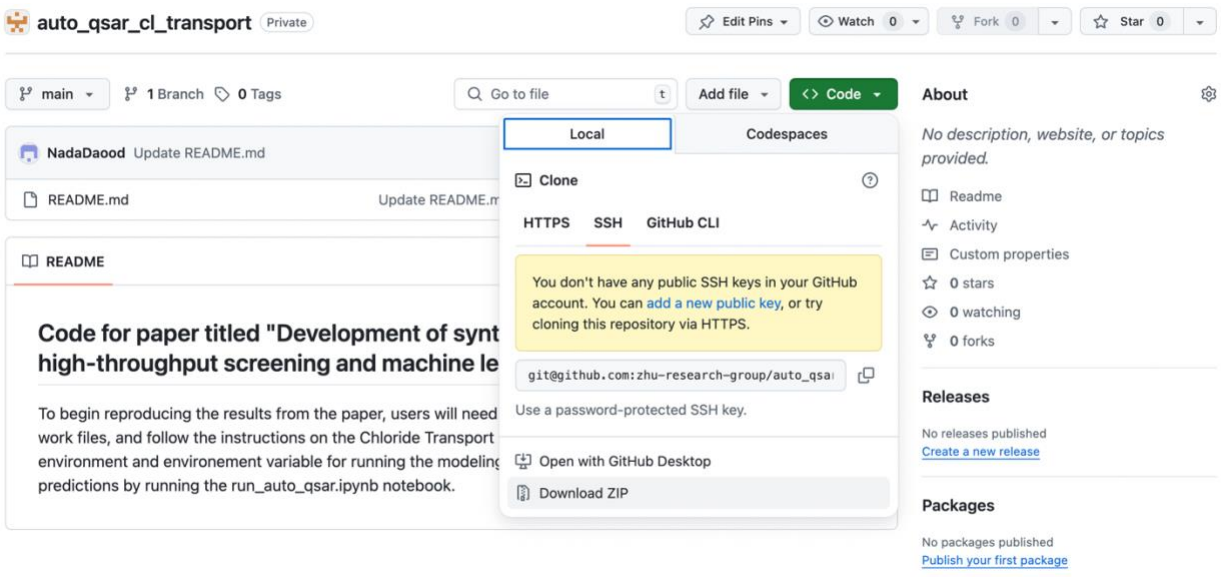
Chloride Transport Models – GitHub and Jupyter Notebook Guide

The following repository reproduces the models and results mentioned in the paper: "Development of synthetic chloride transporters using high-throughput screening and machine learning". This notebook provides a simple guide on how to run the auto_qsar pipeline and generate predictions for external test sets.

The guide to running the pipeline can also be found in more detail in our published book chapter: Ciallella, H. L.; Chung, E.; Russo, D. P.; Zhu, H. Automatic Quantitative Structure–Activity Relationship Modeling to Fill Data Gaps in High-Throughput Screening. In High-Throughput Screening Assays in Toxicology; Zhu, H., Xia, M., Eds.; Methods in Molecular Biology; Humana: New York, NY, 2022; Vol. 2474, pp 169–187. https://doi.org/10.1007/978-1-0716-2213-1_16. Sections from the guide below are taken directly from the book chapter and adjusted slightly to accommodate a few changes in the code for the auto_qsar pipeline.

1. Download the Zip file from our GitHub page.

https://github.com/zhu-research-group/auto_qsar_cl_transport

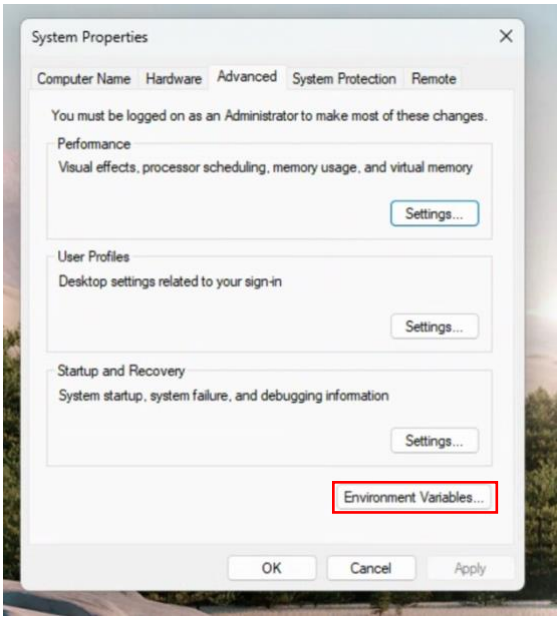


Note: For Windows users, we recommend that users avoid opening or moving the unzipped folder to their OneDrive folder (or any other Cloud-based folder) as that may lead to confusion between the files located on the user's local desktop and the OneDrive account.

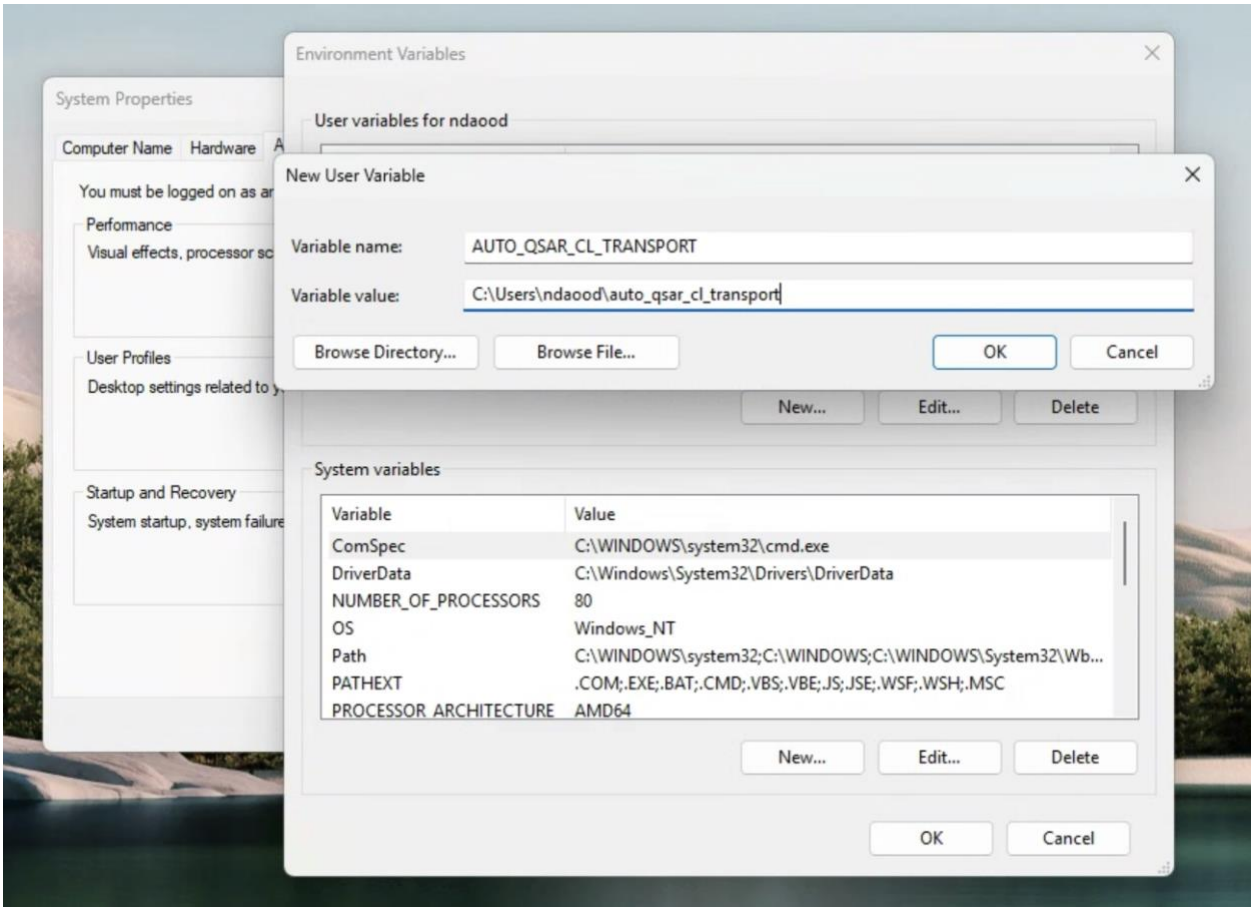
2. Set up a Project Environment Variable.

Windows: First, locate the environment variable panel in either of the following two ways.

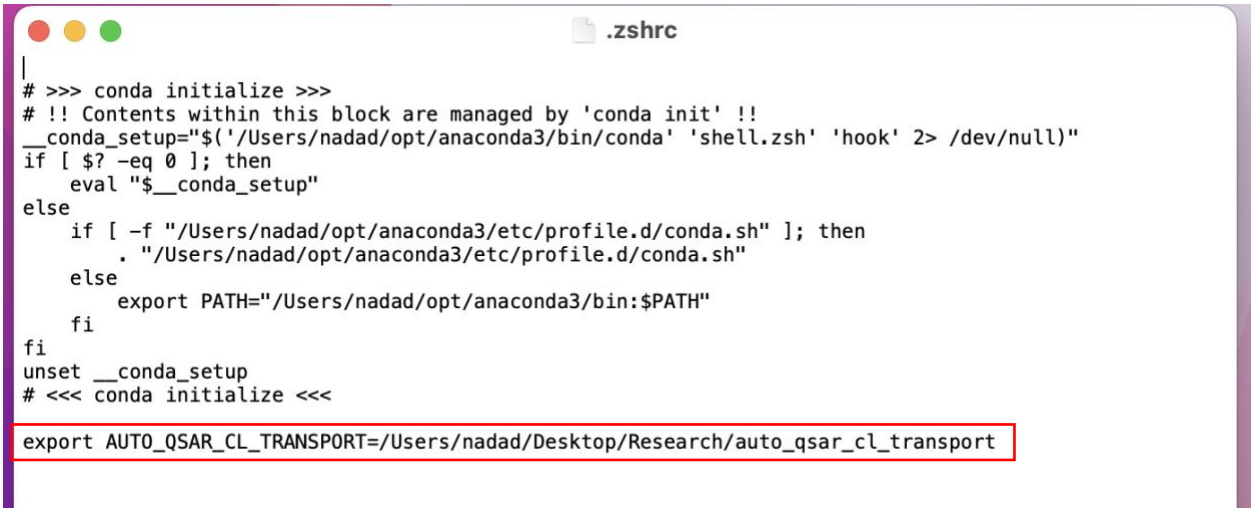
1. Navigate to the Search bar at the bottom of the Windows screen and type “environment variables”. Select “Edit the system environment variables”, then select the “Environment Variables” button at the bottom of the panel in the pop-up window.
2. Another method in Windows 11 is to go to the Windows Control Panel and navigate to “Settings” > “System” > “About” > “Advanced system settings.” Select “Edit the system environment variables”, then select the “Environment Variables” button at the bottom of the panel in the pop-up window.



After clicking the “Environment Variables...” button, the top of the screen displays environment variables defined for only the active user account. The bottom of the screen displays environment variables defined for all user accounts on the computer. This modeling workflow can access either kind of environment variable. Select “New...” under environment variables defined for the active user account. The “Variable name” can be any name without spaces. In this project, the variable name is defined as “AUTO_QSAR_CL_TRANSPORT.” The “Variable value” should be the complete file path to the directory containing the training set files for modeling. Save this environment variable assignment.



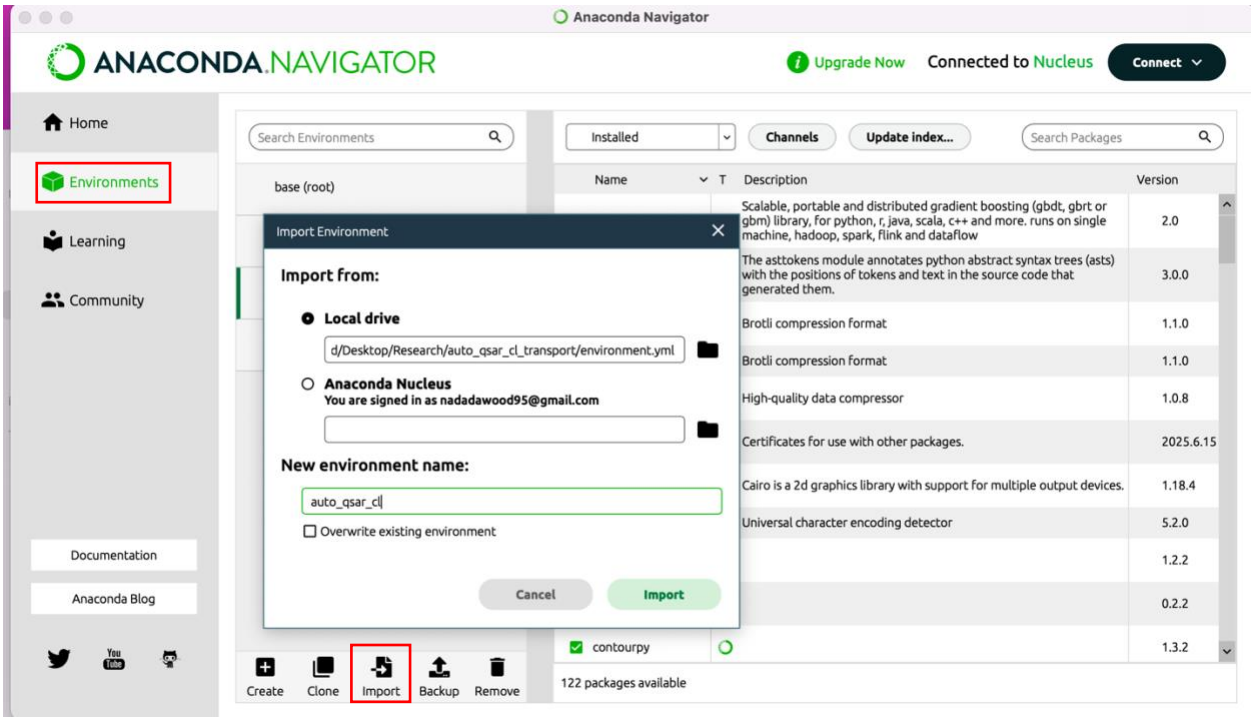
macOS and Linux: Most Linux and macOS installations use Bash as the default shell. However, some newer installations of macOS (e.g., Catalina and Big Sur) use Z shell (Zsh). Users can check which shell their installation uses by opening a Terminal window and entering the command $SHELL. In Terminal, open the active user account's configuration file (i.e., ~/.bash_profile for macOS Bash users, ~/.zshrc for Zsh users, or ~/.bashrc for Linux Bash users) by executing the following command `open -e ~/.zshrc`. Add a line of text into the file defining the environment variable as follows: export VARIABLE="/path/to/file" if using the Bash shell or export VARIABLE='/path/to/file if using Zsh. In this project, the variable name was defined as "AUTO_QSAR_CL_TRANSPORT". Save and exit the configuration file. See example below for macOS.



3. Install the Anaconda® Environment

The Anaconda® distribution of Python installs with many popular data science packages, including those used in this QSAR modeling workflow. An advantage of using Anaconda® is the built-in capability to define environments that load only a program's necessary packages. Further, Anaconda® simplifies the installation of additional field-specific packages and the transfer of environments between computers. This workflow uses a predefined Anaconda® environment that users can easily replicate on their machines. First, download Anaconda® from https://www.anaconda.com/download.

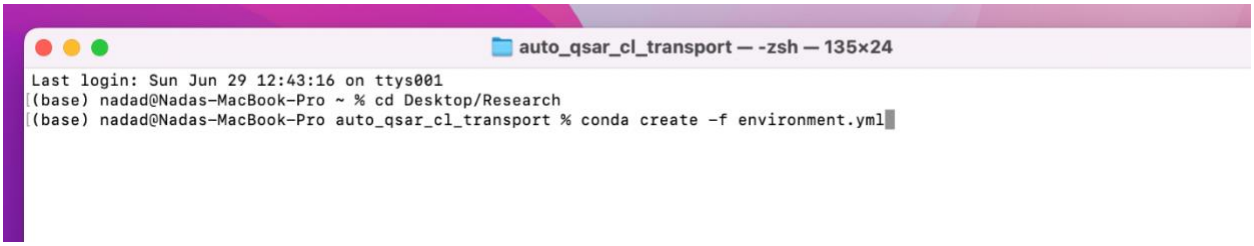
For macOS and Windows After downloading Anaconda®, users can go to the Environments tab on the left panel of the Anaconda® Navigator, then select import at the left bottom section of the screen. At the pop-up window, users can select Local Drive and navigate to the computer directory containing the unzipped workflow files. Select the "environment.yml" file and name the environment "auto_qsar_cl".



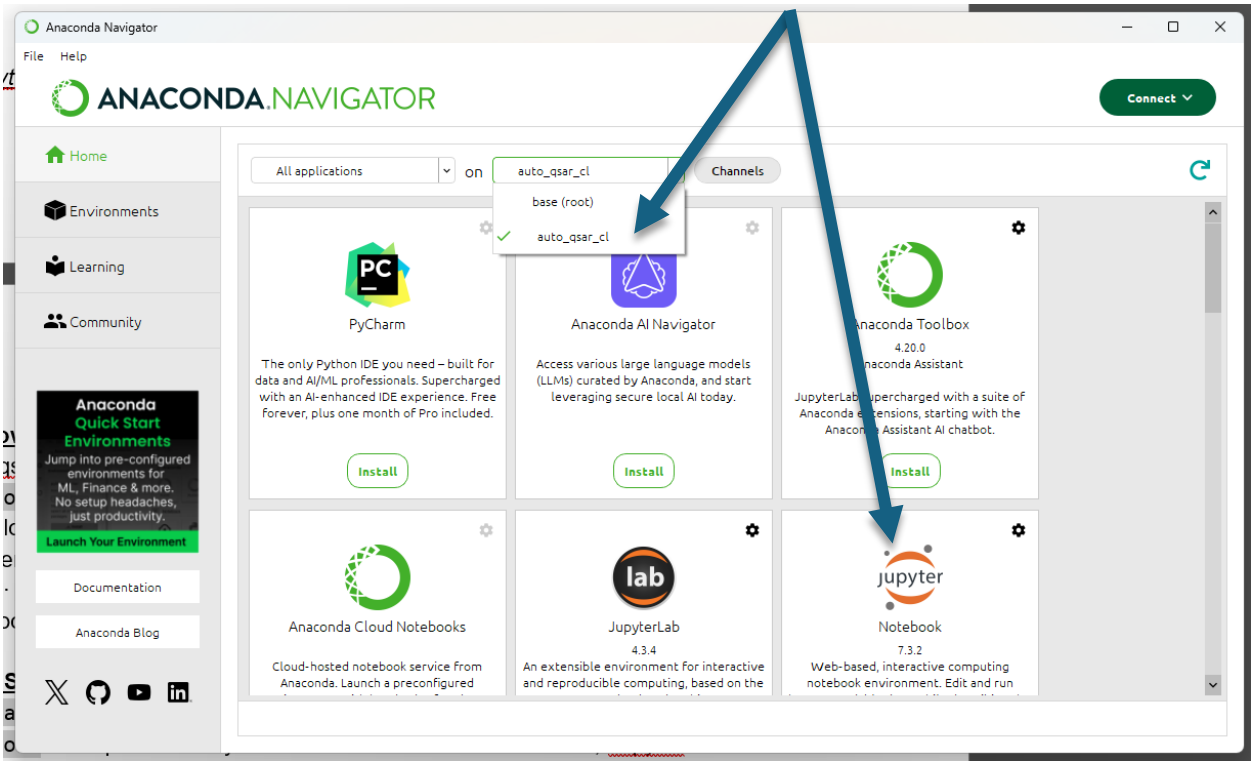
If the above does not work, then follow the instructions below instead:

Windows Navigate to the computer directory containing the unzipped workflow files in Windows File Explorer. Right-click in this directory while holding down the shift key, and select “Open PowerShell window here.” In the PowerShell window, execute the command `conda env create -f environment.yml`

macOS and Linux: Open a new Terminal window, and go to the computer directory containing the unzipped workflow files. Execute the command `conda env create -f environment.yml`



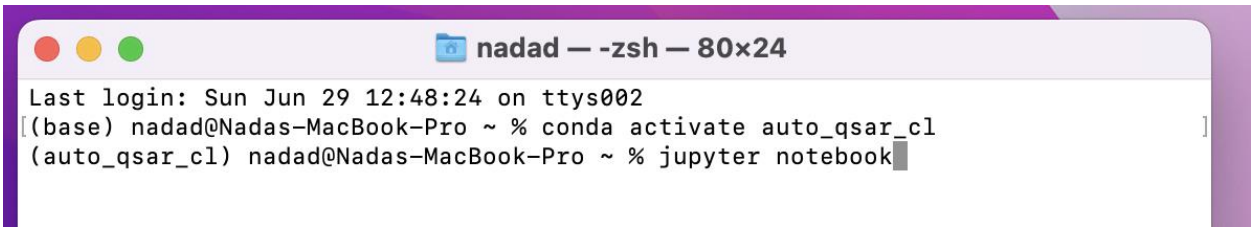
4. Launch Jupyter Notebook from the Anaconda® Navigator software under the auto_qsar_cl environment



If Jupyter Notebook refuses to launch from the Anaconda® Navigator, try the following:

Windows: Open Anaconda® Navigator and launch the Powershell Prompt under the auto_qsar_cl environment. Then, execute the command `conda install jupyter notebook`. This process may take a few minutes. Once the installation finishes, execute the following command: `conda activate auto_qsar_cl`. Then, Jupyter Notebook can be opened by executing the command `jupyter notebook`. The installation step only needs to be done once. When opening the notebooks again, simply type jupyter notebook, and Jupyter Notebook should open in the user's default web browser.

macOS and Linux: open the terminal and execute the following command: `conda activate auto_qsar_cl`. Then, execute the command `conda install jupyter notebook`. This process may take a few minutes. When it's over, Jupyter Notebook can be opened by executing the command `jupyter notebook`. The installation step only needs to be done once. When opening the notebooks again, simply type jupyter notebook, as shown in the screenshot below.



5. Run the run_auto_qsar.ipynb notebook.

In the Jupyter Notebook Home page, navigate to the computer directory containing the unzipped workflow files, and click on ‘run_auto_qsar.ipynb’. This will open the correct Jupyter notebook and you can follow the instructions on the notebook (essentially ‘run all’). If the Jupyter Notebook is giving errors, try opening it from the terminal (as shown above in the previous step) instead of launching it from Anaconda Navigator.