

## Quick guide to using Python preprocessing scripts in SIMCA-online

This document describes how to practically add a script to SIMCA® and use it in SIMCA®-online.

### General information

SIMCA® can since version 16 use and embed Python scripts for preprocessing of data. This lets you morph the dataset for both modelling and predictions (including SIMCA®-Q 16 and SIMCA®-online 16) row by row.

**Note:** SIMCA®-online does not currently support the use of external Python libraries.

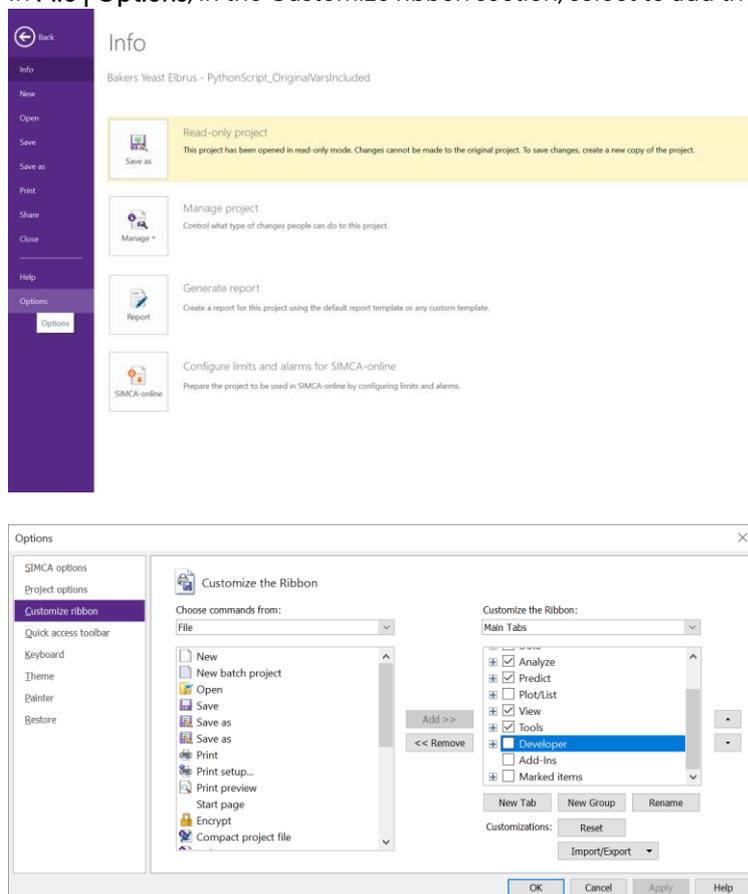
**Note:** Applying a Python script might slow down performance of the SIMCA®-Q and SIMCA®-online systems.

### Creating a Python preprocessing script

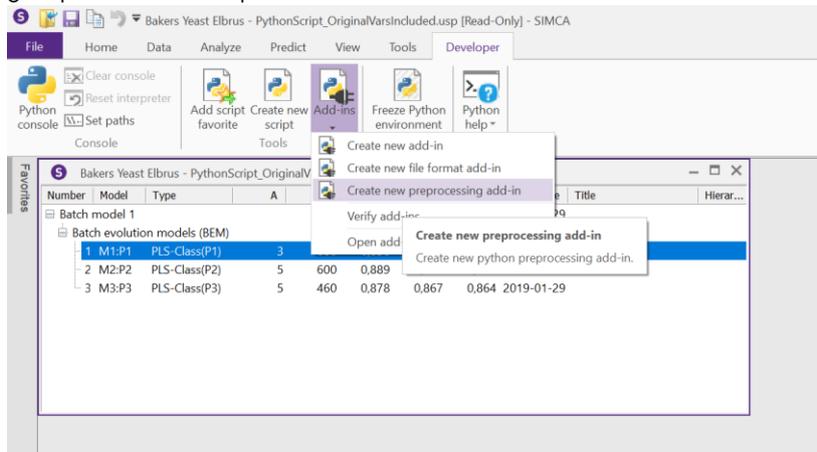
This section describes how to create a Python preprocessing script. The actual Python code is left for you to implement.

1. Add the Development tab to SIMCA® (if it is not there already):

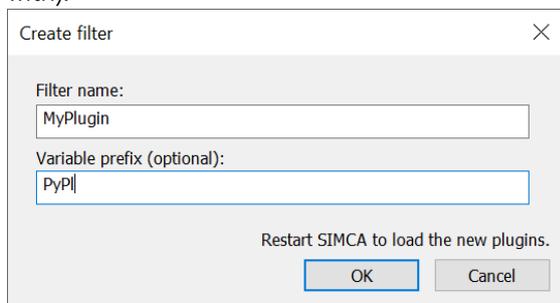
In **File | Options**, in the Customize ribbon section, select to add the Development tab and click **OK**



2. Create a new Python preprocessing add-in by clicking **Add-ins | Create new preprocessing add-in** in the Tools group on the Developer tab



3. Give the script a name, and optionally also a prefix (that all new variables in the new dataset will be prefixed with).



A template is created and automatically opened. This is where you add your Python code

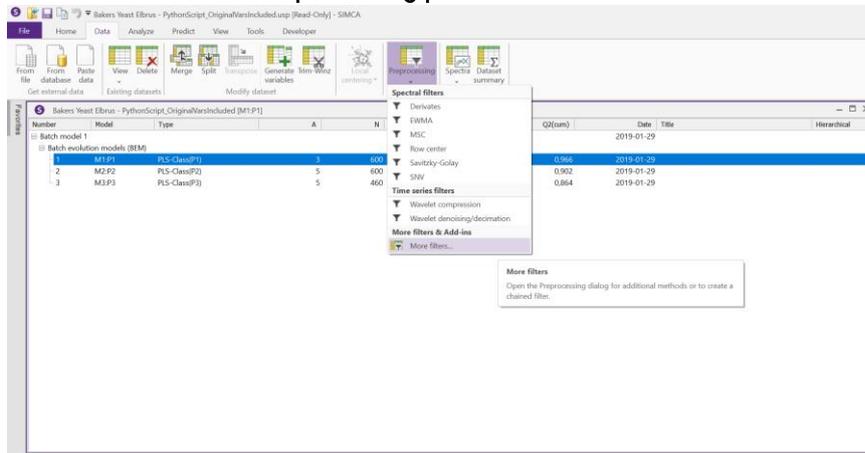


The example function (transform) just takes the square of all values, but this is where you need to enter your own transformation/preprocessing function. The matrix returned must always have the same number of rows as the input data\_collection but other than that anything is possible.

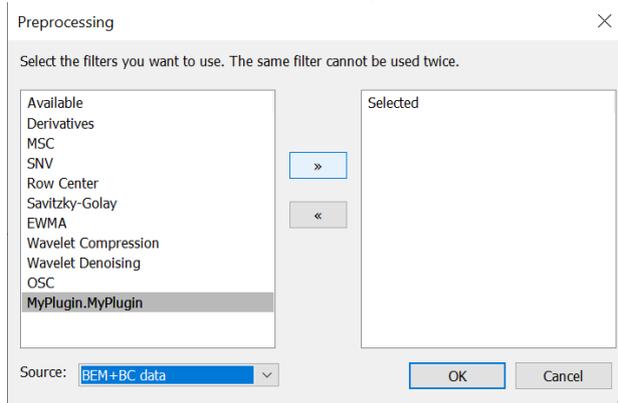
## Applying a preprocessing script

This section describes how a spectral filter and Python preprocessing script can be used in SIMCA®. Note that SIMCA® needs to be restarted to load any new Python add-ins from the add-in folder, and that all preprocessing scripts that you want to use need to be in the add-in folder of SIMCA®.

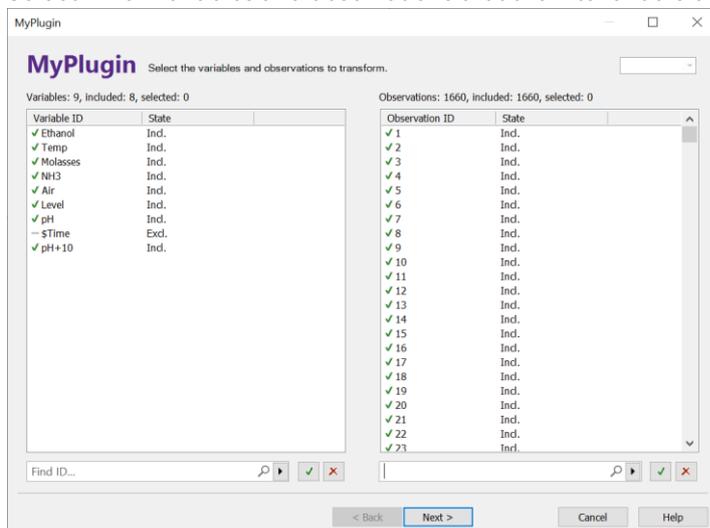
1. On the Data tab, click **Preprocessing | More filters**



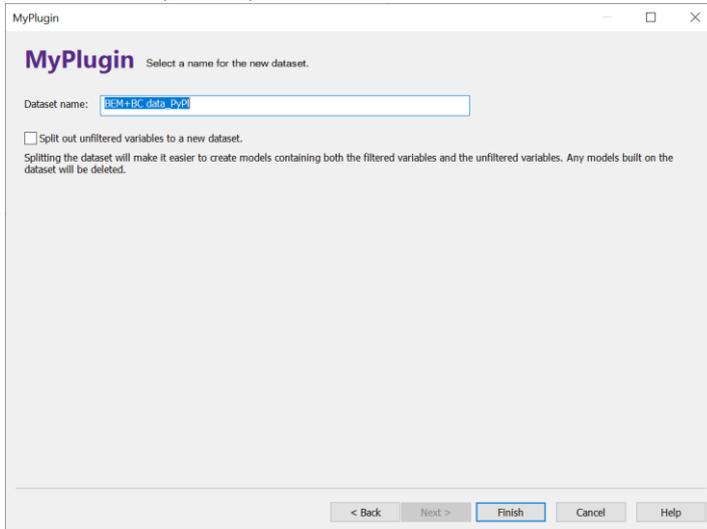
2. Select the Python script (or any of the built-in filters) that you want to use, click >> to add it to the selected filters, select the Source dataset to apply the filtering to, and click **OK** to continue with other settings. Note that several filters can be applied in a chain



3. Select which variables and observations that the filter should affect and click **Next**



4. Give the new (filtered) dataset a name and click **Finish**



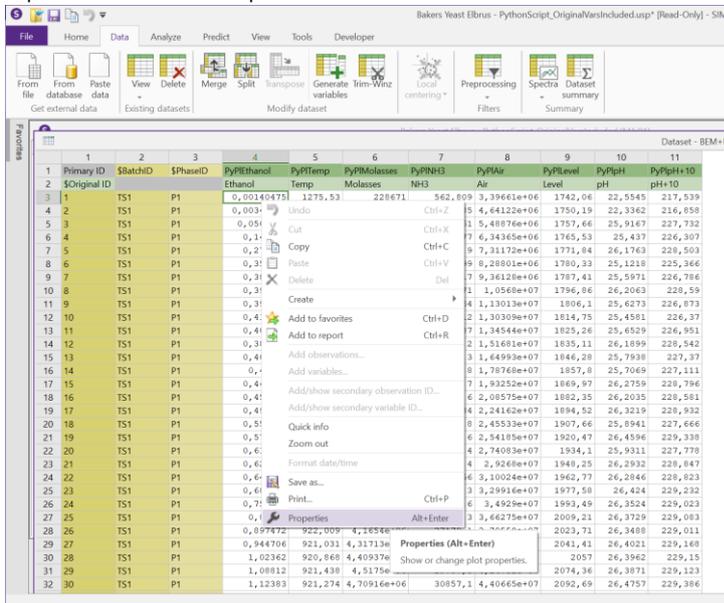
The screenshot shows a dialog box titled "MyPlugin" with a close button (X) in the top right corner. The main heading is "MyPlugin" followed by the instruction "Select a name for the new dataset." Below this, there is a text input field labeled "Dataset name:" containing the text "SEM+BC\_data\_DVP". Underneath the input field is a checkbox labeled "Split out unfiltered variables to a new dataset." with a small explanatory text below it: "Splitting the dataset will make it easier to create models containing both the filtered variables and the unfiltered variables. Any models built on the dataset will be deleted." At the bottom of the dialog, there are five buttons: "< Back", "Next >", "Finish" (which is highlighted with a blue border), "Cancel", and "Help".

## Viewing an existing embedded Python preprocessing script

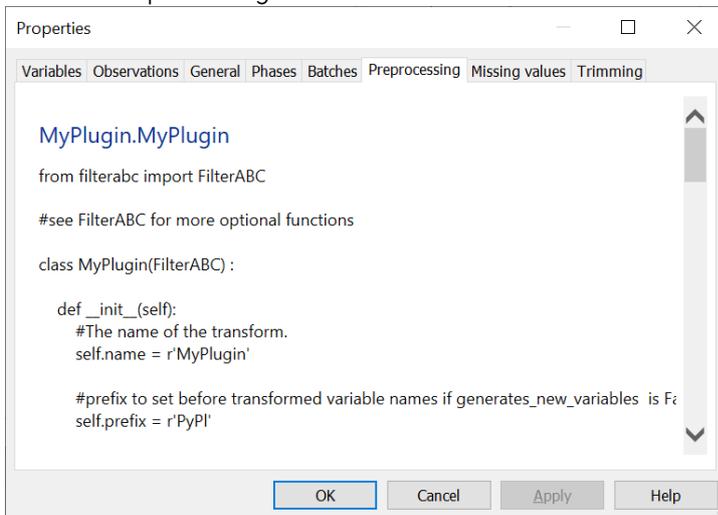
This section describes how to view the embedded Python script for a specific dataset in SIMCA®.

To view the existing Python preprocessing script that is already embedded in a dataset:

1. Open the dataset Properties:



2. View the Preprocessing tab:

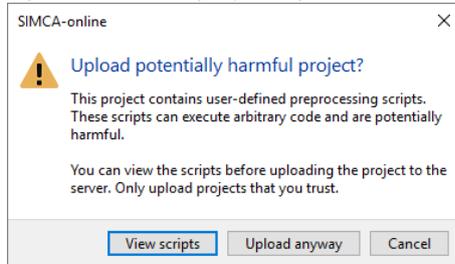


## Using a SIMCA project with an embedded Python preprocessing script in SIMCA®-online

This section describes using a SIMCA® project with an embedded Python script in SIMCA®-online.

Note that since Python scripts can potentially contain dangerous code, any project with existing Python scripts needs to be approved before it can be used in the products.

1. Upload SIMCA® project opens a warning-message:



To see the Python script content, click **View scripts**. After reviewing the script, provided you are content, click **Upload anyway** to move forward

2. Step through the project configuration as with any other project.
3. Display your new/filtered variables in the charts