

## **Filter Descriptors**

Shaillay Kumar Dogra  
Scientific Editor – QSAR World  
[editor@qsarworld.com](mailto:editor@qsarworld.com)

### **Notes:**

1. This Jython script works in Sarchitect Designer version 2.2 & 2.3
2. Learn about Sarchitect Designer – <http://www.strandls.com/sarchitect/index.html>
3. Get Sarchitect – <http://www.strandls.com/sarchitect/freetrial.php>

***The actual script follows this discussion. It is also accessible directly from the webpage in .py format.***

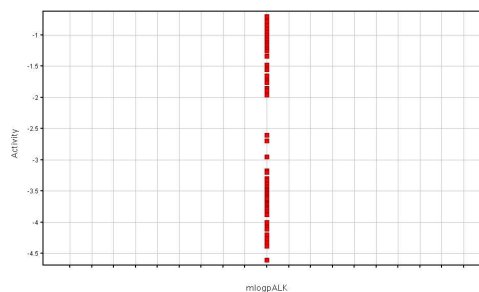
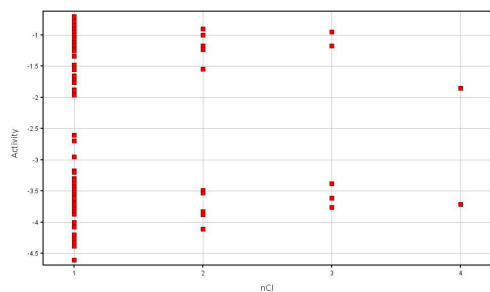
### **Discussion:**

In order to fit a QSAR model between a given endpoint and some selected descriptors, there should be some correspondence between the variation in the endpoint and the descriptors under consideration. In other words, if the descriptor does not exhibit much variation vis-à-vis the endpoint it may be of little significance for modeling. This is particularly true for the regression modeling scenario.

The script provided here filters out such descriptors (that show low variation). Specifically, the user is asked to set a cutoff value based on which a given descriptor under consideration is either retained or rejected. This cutoff pertains to the number of distinct values a descriptor should exhibit. The default is dynamically set as 10% of the total data points and this is the value that is prompted by the script in a dialog-box. The user can change the cutoff by providing some other number as the minimum number of distinct values that a given descriptor should exhibit for it to be retained.

Output is a child dataset named as "Filtered Set" that contains descriptors that passed the filtering criterion based on the cutoff for minimum number of distinct values. Also, the descriptors that were filtered out are shown in a scatter plot where the x-axis has the list of descriptors that were rejected and the y-axis is the endpoint. This plot allows one to visually inspect the descriptors that were removed and the kind of variation (or rather lack of it) that they exhibited vis-à-vis the endpoint.

Say, there are 100 compounds in the dataset. The cutoff automatically gets defined as 10. If the user now changes this to 5 then all the descriptors that have less than 5 distinct values are dropped. Two example plots of such dropped descriptors are shown.



**Cite this as:**

Dogra, Shaillay K., "Script for filtering descriptors" from QSARWorld – free online resource for QSAR modeling. <http://www.qsarworld.com/virtual-workshop.php>

```
##
##
## Sarchitect designer 2.3 script to filter descriptors that have less
than certain number of distinct values
##
## Shaillay Dogra
## 25 July 2007
## editor@qsarworld.com
##
##
## User sets a cutoff of how many minimum distinct values should exist
in a given descriptor column
## Based on the cutoff the descriptor is retained or rejected
## The descriptors that were filtered out are displayed in a scatter
plot
## A subset is created containing those descriptors that passed the
filtering criterion
##
##

import script
from script.dataset import *
from script.algorithm import *
from script.project import *
from script.view import *
from script.omega import createComponent, showDialog
from javax.swing import *
from math import *

##-----
## DEFINE CHECKDATA
def checkdata(dataset):
    indices_continuous =
DatasetUtil.getContinuousColumnIndices(dataset)
    if(indices_continuous.getSize()==0):
        parent=script.tool.getTool().getFrame()
        mesg = "No Descriptor Data"

        JOptionPane.showMessageDialog(parent,mesg,"ERROR!",JOptionPane.IN
FORMATION_MESSAGE)
        return 0
    else:
        return 1
##-----

## DEFINE DISTINCEVALUES
def distinctvalues(column):
    from java.util import HashSet
    set = HashSet()
    for rowIndex in range(column.getSize()):
        set.add(column.get(rowIndex))
    values = set.size()
    return values

##-----
```

```
##
## DEFINE MAIN
##

def main(dataset):

    ## Get descriptor columns, assumption: continuous and unmarked
    columns
    indices_continuous =
DatasetUtil.getContinuousColumnIndices(dataset)
    indices_nm_continuous =
script.project.removeMarkedColumns(dataset, indices_continuous)
    columnList = indices_nm_continuous
    #print columnList

    ## Get endpoint column
    classlabelcolumnIndex = indices_continuous.get(0)
    classlabelCol = DatasetUtil.getMarkedColumns(dataset, "Endpoint")
    for col in classlabelCol:
        if(not col.isCategorical()):
            classlabelcolumnIndex = dataset.indexOf(col)
        break
    #print "Endpoint:", dataset[classlabelcolumnIndex]

    total_points = dataset.getRowCount()
    dynamic_default = total_points/10

    ## Ask user for minimum number of distance values to be used as
    cut-off
    p = createComponent(type="float", id="name", description="Minimum
Distinct Values?", value=dynamic_default)
    cutoff=showDialog(p)
    #print cutoff

    passlist = []
    faillist = []

    ## Check number of distinct values
    for i in range(columnList.getSize()):
        idx = columnList.get(i)
        col = dataset[idx]
        howmany = distinctvalues(col)
        #print howmany

        if (howmany >= cutoff):
            #print col, howmany
            passlist.append(col)
        else: faillist.append(col)

    #print passlist, " >>>-<<< ", faillist

    ## Show a scatterplot containing columns that failed the
    filtering above
```

```
rowIndices = [i for i in range(dataset.getRowCount())]
colIndices = faillist
endpoint = dataset[classlabelcolumnIndex]
colIndices.append(endpoint)
endIdx = len(colIndices)-1 # since count starts from zero
tmpdataset = script.dataset.createDataset("Failed Set",
colIndices)
script.view.ScatterPlot(dataset=tmpdataset, yaxis=endIdx).show()

## Define a new child dataset containing columns that passed the
filtering above

rowIndices = [i for i in range(dataset.getRowCount())]
colIndices = []
script.project.addSubsetChild(rowIndices, colIndices ,
name="Filtered Set", addMarkedColumns=1)
subset=script.project.getActiveDataset()

for col in passlist:
    subset.addColumn(col)
script.view.Table(dataset=subset).show()

##-----

## Call main
dataset = script.project.getActiveDataset()
if (checkdata(dataset)!=0):
    main(dataset)

## Report completion
parent=script.tool.getTool().getFrame()
mesg = "Done With Script Execution."
JOptionPane.showMessageDialog(parent,mesg,"STATUS!",JOptionPane.I
NFORMATION_MESSAGE)

##
## END
##
```

---

End of Document