

## **Correlation Based Filtering**

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 and 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:**

This script filters off descriptors whose correlation with the endpoint is less than the user-defined cutoff. The correlation of a given descriptor with the endpoint is calculated and the descriptor is retained if the absolute correlation value is more than the user-defined cutoff.

As an input, it prompts the user to provide the cutoff value of correlation. A subset folder is created containing all the marked columns and those descriptors that satisfied the correlation based filtering. This subset is named as "Correlation Set".

### **Cite this as:**

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

---

```
##
##
##
## Sarchitect designer script to pick descriptors based on correlation
##
## Shaillay Dogra
## 24 July 2007
## editor@qsarworld.com
##
## Correlation of each descriptor with the end-point is calculated
## User sets a cut-off
## Based on the cut-off the descriptor is retained or rejected
##
##

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 SUM
def sum(list):
    total = 0.00
    for val in list:
        total = total + val
    return total
##-----

##DEFINE SUM OF SQUARES
def sumofsquares(list):
    total = 0.00
    for val in list:
        total = total + (val*val)
    return total
##-----

## DEFINE 'MEAN' FUNCTION
def mean(list):
    count = len(list)
    total = sum(list)
    mean = (total/count)
    return mean
##-----

##DEFINE 'COPRODUCT' FUNCTION
def coproduct(list1, list2):
    n = len(list1)
    coproduct = [ ]
    for idx in range(n):
        product = list1[idx] * list2[idx]
```

```

        coproduct.append(product)
    return coproduct
##-----

##DEFINE CORRELATION
def correlation(list1, list2):
    n = len(list1)

    num = (n*sum(coproduct(list1, list2))) -
    (sum(list1)*sum(list2))
    dnm = sqrt((n*sumofsquares(list1)) - (sum(list1)**2)) *
    sqrt((n*sumofsquares(list2)) - (sum(list2)**2))

    if (dnm!=0.0):
        corr = num/dnm
    else: corr = Float.MAX_VALUE
    return corr
##-----

## 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 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]

```

```

    ## Ask user for correlation cut-off
    p = createComponent(type="float", id="name",
description="Correlation Cut-off?", value="0.80")
    cutoff=showDialog(p)
    #print '%.2f'% cutoff

    ## Get endpoint values in a list
    endpoint = []
    for val in dataset[classlabelcolumnIndex]:
        endpoint.append(val)
    #print endpoint

    passlist = []

    ## Compute correlaton b/w a descriptor and endpoint
    for i in range(columnList.getSize()):
        idx = columnList.get(i)
        col = dataset[idx]
        collist = []
        for val in col:
            collist.append(val)

        corr = correlation(endpoint, collist)
        abscorr = abs(corr)
        #print '%.2f'% abscorr

        if (abscorr > cutoff and abscorr <=1.0): #latter check
because we set corr = Float.MAX_VALUE when division be zero happens
            #print col, '%.2f'% abscorr
            passlist.append(col)

    #print passlist

    ## Define a new child dataset

    rowIndices = [i for i in range(dataset.getRowCount())]
    colIndices = []
    script.project.addSubsetChild(rowIndices, colIndices ,
name="Correlation 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

---