

## Tanimoto Coefficient

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. 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:

Determining if two compounds are similar to each other or not is an important problem in chemistry, especially in context of QSAR modeling, given the underlying assumption of the similarity principle [sim-principle].

Various similarity metric exist that return a score indicating the level of similarity between two molecules under comparison [chem-sim]. Frequently used metrics are simple distance measures such as [Hamming](#) and [Euclidean](#) distance, and association coefficients such as [Tanimoto](#), [Dice](#) and Cosine coefficients.

A simple count of shared features (common fragment substructures) can be a measure of chemical distance when used in some similarity coefficient. Dictionaries of predefined structural fragments, such as MDL Information Systems' [MACCS](#) keys, are used to identify features contained in a molecule. The structural fragments or features that are present in the given molecule are turned ON (set as 1) and the ones that are absent are kept OFF (set as 0). Thus, for each molecule one ends up having a string containing 1s and 0s (bit string).

Once the molecules have been represented by such bit-strings Tanimoto coefficient can be used as a measure to assess similarity. Let's say, we are comparing two molecules A and B. If  $N_A$  is number of features (ON bits) in A,  $N_B$  is the number of features (ON bits) in B, and  $N_{AB}$  is the number of features (ON bits) common to both A and B, then, Tanimoto coefficient simply is:

$$T = N_{AB} / N_A + N_B - N_{AB}$$

Note that the OFF bits do not determine the similarity. In other words, if some molecular features are absent in both molecules then that is not taken as an indication of similarity between the two [fingerprint-sim].

The script provided here computes Tanimoto coefficient between all pairs of compounds thus displaying an  $n \times n$  matrix. Compounds that have Tanimoto coefficient values  $> 0.85$  are generally considered similar to each other.

As an input it requires a 'Structure' column and an 'Identifier' column, accessing which from the underlying spreadsheet is hard-coded in lines 57 and 81 of the script. In case of error, check your column names and change the script accordingly.

MACCS key computation is called from the backend and not done in the script. Tanimoto coefficient is computed within the script (though these coefficients too can be accessed from the backend). As an extension of this script I would be implementing many other coefficients (not computed at the backend) and computing Tanimoto coefficient seemed like a good starting point.

**References:**

[sim-principle] Dogra, Shaillay K., "Similarity Principle" From QSARWorld – A Strand Life Sciences Web Resource. <http://www.qsarworld.com/insilico-chemistry-similarity-principle.php>

[chem-sim] Dogra, Shaillay K., "Chemical Similarity." From QSARWorld – A Strand Life Sciences Web Resource. <http://www.qsarworld.com/insilico-chemistry-chemical-similarity.php>

[fingerprint-sim] Dogra, Shaillay K., "Fingerprint-based Similarity." From QSARWorld – A Strand Life Sciences Web Resource. <http://www.qsarworld.com/insilico-chemistry-fingerprint-based-similarity>.

**Cite this as:**

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

---

```

##
##
## sarchitect designer 2.2 script to compute Tanimoto coefficients (all ## pairs within dataset)
##
## Shaillay Kumar Dogra
## editor@qsarworld.com
## Feb 20, 2007
##
## INPUT: Data with 'Identifier' and 'Structure' column
##
## MACCS key computation is called from backend algorithm
##
##
## OUTPUT: n X n matrix containing Tanimoto coefficient values
##
## TO DO: not compute half of the n X n matrix
##

```

```

from com.strandgenomics.chem.molalgorithms.util import *
from java.lang import Float
from script.omega import createComponent, showDialog
from javax.swing import *

```

```

##-----
## COMPUTE TANIMOTO COEFFICIENT
def tanimoto(key1, key2):
    nA = 0.0  ## no. of features ON in molecule A
    nB = 0.0  ## no. of features ON in molecule B
    nAB = 0.0 ## no. of features ON in both molecule A & B

    for i in range(len(key1)):
        if (key1[i]!=0): ## feature is ON in A
            nA = nA + 1
            if (key2[i]!=0): ## AND feature is ON in B also; count doesn't matter
                #print key1[i], key2[i]
                nAB = nAB + 1

        if (key2[i]!=0): ## feature is ON in B
            nB = nB + 1

    if ((nA + nB - nAB)!=0):
        tanimoto = nAB / (nA + nB - nAB)
    else: tanimoto = Float.MAX_VALUE

    return tanimoto

```

```

#-----

```

```

## MAIN

```

```

dataset = script.project.getActiveDataset()

```

```
strCol = dataset.getColumn('Structure')

result = script.algorithm.ComputeMACCSKey(structure=strCol).execute(displayResult=0)
keys = result['maccskeymatrix']
#print len(keys), keys[0], keys[1]

coeff_cols = []
total_rows = dataset.getRowCount()
for i in range(total_rows):
    row_coeff = []
    j = 0
    while (j < total_rows):
        coeff = tanimoto(keys[i], keys[j])
        row_coeff.append(coeff)
        j = j+1

    new_col = script.dataset.createFloatColumn(str(dataset[0][i]), row_coeff)
    coeff_cols.append(new_col)

## end of for loop

## create metric dataset, launch view
id_col = dataset.getColumn('Identifier')
coeffset = script.dataset.createDataset("Coefficient", [id_col])
for i in range(len(coeff_cols)):
    coeffset.addColumn(coeff_cols[i])

view = script.view.Table(Title='Tanimoto Coefficients', dataset=coeffset, rowHeight=40)
view.__state__['enableExportColumns'] = 1
view.show()

## report completion
parent=script.tool.getTool().getFrame()
mesg = "Done With Script Execution."
JOptionPane.showMessageDialog(parent,mesg,"STATUS!",JOptionPane.INFORMATION_MESS
AGE)

##
## END
##
```

---

End of Document