Modified K-Medoids

A Clustering Tool

NANOBRIDGES
-A Collaborative Project

UNIWERSYTET GDAŃSKI

Marie Curie Actions
EUROPEAN UNION

THE AUTHORS ARE GRATEFUL FOR THE FINANCIAL SUPPORT FROM THE EUROPEAN COMMISSION THROUGH THE MARIE CURIE **IRSES** PROGRAM, NANOBRIDGES PROJECT (FP7-PEOPLE-2011-IRSES, GRANT AGREEMENT NUMBER 295128).

MODIFIED K-MEDOIDS CLUSTERING TOOL

Clustering is the process of grouping a set of objects into clusters, so that the objects within a cluster are similar to each other but are dissimilar to objects in other clusters. K-means clustering and partitioning around medoids (PAM) are well known techniques for performing non-hierarchical clustering. K-means clustering iteratively finds the *k* centroids and assigns every object to the nearest centroid, where the coordinate of each centroid is the mean of the coordinates of the objects in the cluster. Unfortunately, *K*-means clustering is known to be sensitive to the outliers although it is quite efficient in terms of the computational time. For this reason, K-medoids clustering are sometimes used, where representative objects called medoids are considered instead of centroids. Because it is based on the most centrally located object in a cluster, it is less sensitive to outliers in comparison with the K-means clustering. Among many algorithms for K-medoids clustering, PAM proposed by *Kaufman and Rousseeuw* is known to be most powerful. However, PAM has a drawback that it works inefficiently for a large data set due to its time complexity.

Further, "modified K-Medoid" is a simple and fast algorithm for *k*-medoids clustering. This method tends to select *k* most middle objects as initial medoids [1].

Proposed modified K-medoids algorithm **[1]**

Suppose that *n* objects having *p* variables each should be grouped into *k* ($k < n$) clusters, where, *k* is assumed to be provided by user. Let us define, j^{th} variable of object *i* as X_{ij} ($i = 1, \ldots, n$; $j = 1, \ldots, p$). The Euclidean distance will be used as a dissimilarity measure. The Euclidean distance between object *i* and object *j* is given by *equation (1)*

$$d_{ij} = \sqrt{\sum_{a=1}^{p} (X_{ia} - X_{ja})^2} \qquad i=1, \ldots, n; j=1, \ldots, n \qquad \ldots (1)$$

The proposed algorithm is composed of the following three steps.

Step 1: *(Select initial medoids)*

1-1. Calculate the distance between every pair of all objects based on the chosen dissimilarity measure (Euclidean distance in this case).

1-2. Calculate v_j for object *j* as shown in *equation (2)*

$$v_j = \sum_{i=1}^{n} \frac{d_{ij}}{\sum_{l=1}^{n} d_{il}}, \qquad j=1, \ldots, n \qquad \ldots(2)$$

1-3. Sort v_j's in ascending order. Select k objects having the first *k* smallest values as initial medoids.

1-4. Obtain the initial cluster result by assigning each object to the nearest medoid.

1-5. Calculate the sum of distances from all objects to their medoids.

Step 2: *(Update medoids)*

Find a new medoid of each cluster, which is the object minimizing the total distance to other objects in its cluster. Update the current medoid in each cluster by replacing with the new medoid.

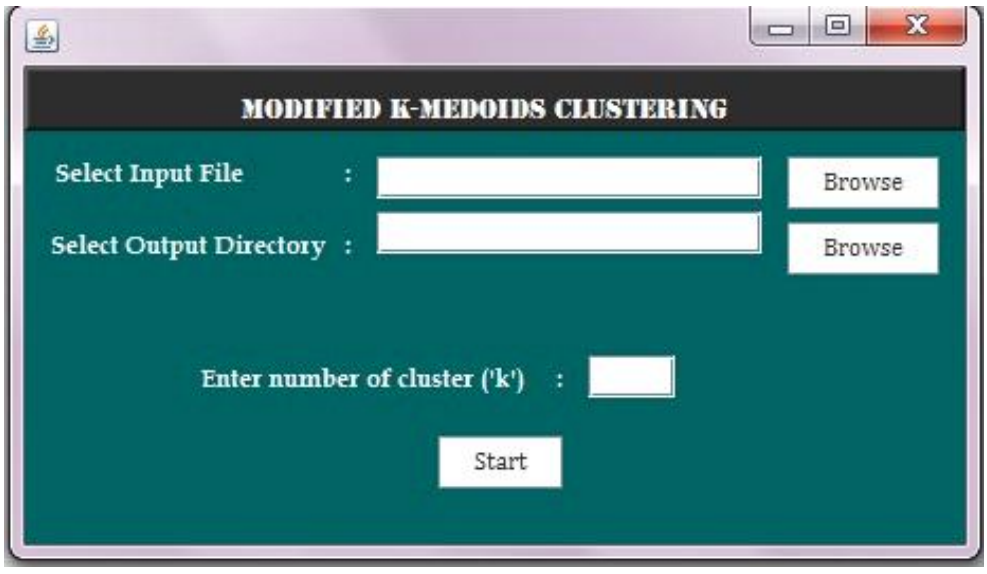
Step 3: *(Assign objects to medoids)*

3-1. Assign each object to the nearest medoid and obtain the cluster result.

3-2. Calculate the sum of distance from all objects to their medoids. If the sum is equal to the previous one, then stop the algorithm. Otherwise, go back to the Step 2.

The above algorithm is a local heuristic that runs just like K-means clustering when updating the medoids. In Step 1, we proposed a method of choosing the initial medoids. This method tends to select *k* most middle objects as initial medoids. The performance of the algorithm may vary according to the method of selecting the initial medoids.

Snapshot 1: Modified K-Medoids



Modified K-Medoids Program Folder

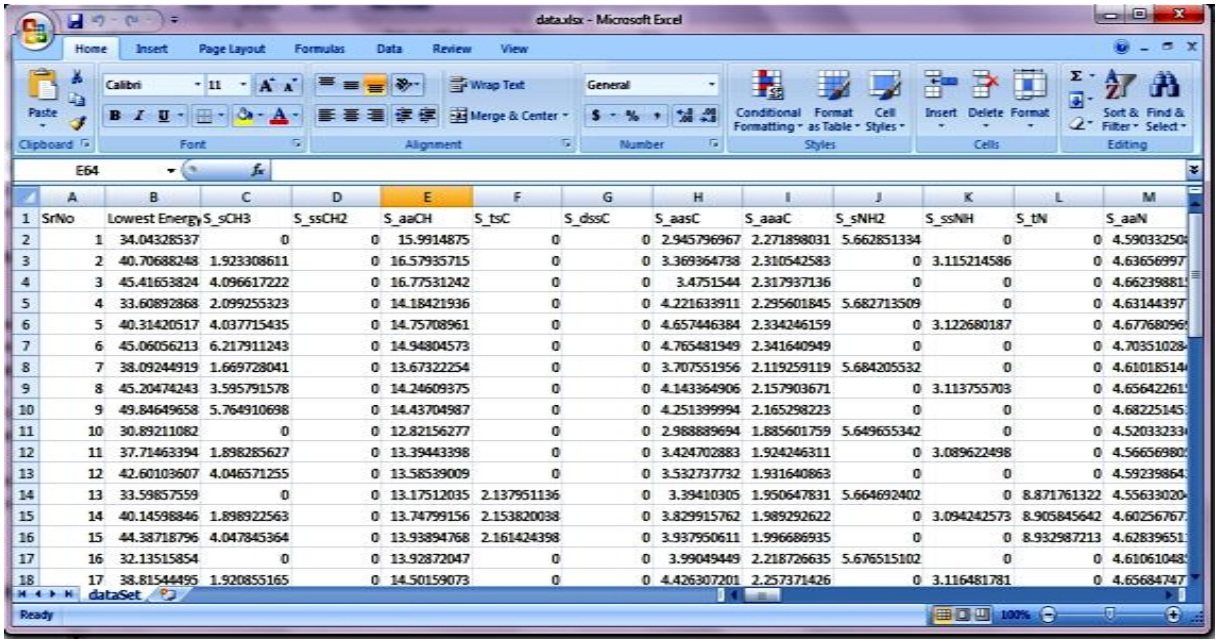
The program folder consists of three folders "**Data**", "**Lib**" and "**Output**". For convenience, user may keep input file in "**Data**" folder and may save output files in "**Output**" folder, since by default, clicking on the browse button will open these folders.

"**Lib**" folder consists of library files required for running the program. Hence do not move or delete or rename these files.

Input file format

Three different file types are allowed *i.e.* ***xlsx , xls*** and ***csv*** as input file. The input file (*see snapshot 2*) should consist of serial number column (*first column*), and descriptor values columns (*subsequent columns*) for each object/compound. The format in which this information should be placed in the input file is as follows:

Snapshot 2



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	SrNo	Lowest Energy	S_sCH3	S_ssCH2	S_aaCH	S_tsC	S_dssC	S_aasC	S_aaaC	S_sNH2	S_ssNH	S_tN	S_aaN
2	1	34.04328537	0	0	15.9914875	0	0	2.945796967	2.271898031	5.662851334	0	0	4.590332508
3	2	40.70688248	1.923308611	0	16.57935715	0	0	3.369364738	2.310542583	0	3.115214586	0	4.636569971
4	3	45.41653824	4.096617222	0	16.77531242	0	0	3.4751544	2.317937136	0	0	0	4.662398811
5	4	33.60892868	2.099255323	0	14.18421936	0	0	4.221633911	2.295601845	5.682713509	0	0	4.631443971
6	5	40.31420517	4.037715435	0	14.75708961	0	0	4.657446384	2.334246159	0	3.122680187	0	4.677680961
7	6	45.06056213	6.217911243	0	14.94804573	0	0	4.765481949	2.341640949	0	0	0	4.703510284
8	7	38.09244919	1.669728041	0	13.67322254	0	0	3.707551956	2.119259119	5.684205532	0	0	4.610185144
9	8	45.20474243	3.595791578	0	14.24609375	0	0	4.143364906	2.157903671	0	3.113755703	0	4.656422615
10	9	49.84649658	5.764910698	0	14.43704987	0	0	4.251399994	2.165298223	0	0	0	4.682251453
11	10	30.89211082	0	0	12.82156277	0	0	2.988889694	1.885601759	5.649655342	0	0	4.520332336
12	11	37.71463394	1.898285627	0	13.39443398	0	0	3.424702883	1.924246311	0	3.089622498	0	4.566569802
13	12	42.60103607	4.046571255	0	13.58539009	0	0	3.532737732	1.931640863	0	0	0	4.592398643
14	13	33.59857559	0	0	13.17512035	2.137951136	0	3.39410305	1.950647831	5.664692402	0	8.871761322	4.556330203
15	14	40.14598846	1.898922563	0	13.74799156	2.153820038	0	3.829915762	1.989292622	0	3.094242573	8.905845642	4.602567677
16	15	44.38718796	4.047845364	0	13.93894768	2.161424398	0	3.937950611	1.996686935	0	0	8.932987213	4.628396511
17	16	32.13515854	0	0	13.92872047	0	0	3.99049449	2.218726635	5.676515102	0	0	4.610610483
18	17	38.81544495	1.920855165	0	14.50159073	0	0	4.426307201	2.257371426	0	3.116481781	0	4.656847477

First Row: Header *i.e.* name for each column, for instances, descriptor names. It can be numerical, alphabet or alphanumerical in nature.

First column: Serial number/ Compound number. It should be numerical and not alphabet or alphanumerical in nature.

Subsequent columns: Property/Independent variables/Descriptor values; each column will consist of descriptor values for all the compounds/objects. These values should be numerical values and not alphabets or alphanumerical values.

Note: If activity column (as last column) is present in the input file it will be considered as one of the descriptor.

How to run the program

It's simple! Just click/double click on the jar file (ModifiedKMedoid.*jar*) present in the '***ModifiedKMedoid***' folder. A window will open as shown in *Snapshot 1*, with few queries, which a user has to fill before clicking on '*Start*' button to run the program.

"***Select Query Input File***": Click on 'browse' button to select the input file. By default, it will open the "Data" folder present in '*ModifiedKMedoid*' folder. So for convenience, user can keep the input file in the "Data" folder.

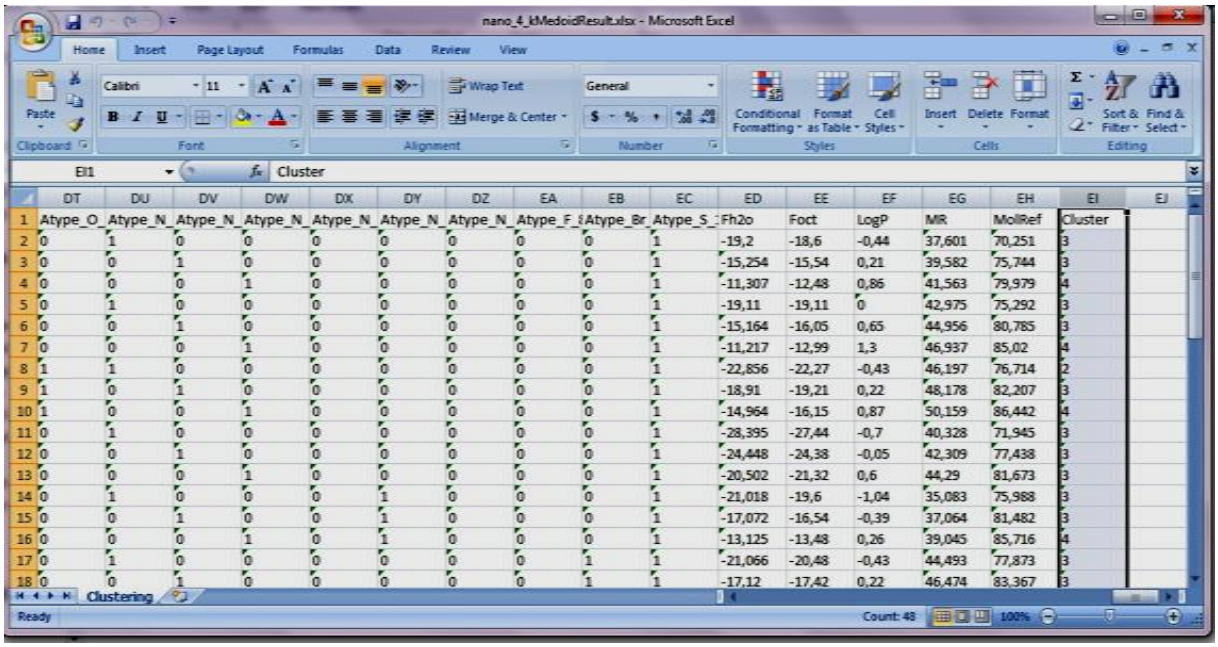
"***Select Output Directory***": Click on 'browse' button to select the destination/output file directory and define output file name. By default, it will open the "*Output*" folder present in '*ModifiedKMedoid*' folder. So for convenience, user can save the output files in the "Output" folder.

"Enter number of cluster 'k'": Enter a value, which will correspond to the number of clusters formed.

Output

The output consist of two files *i.e.* a excel file (*snapshot 3*) and a text file (*snapshot 4*). The content of these files are discussed below.

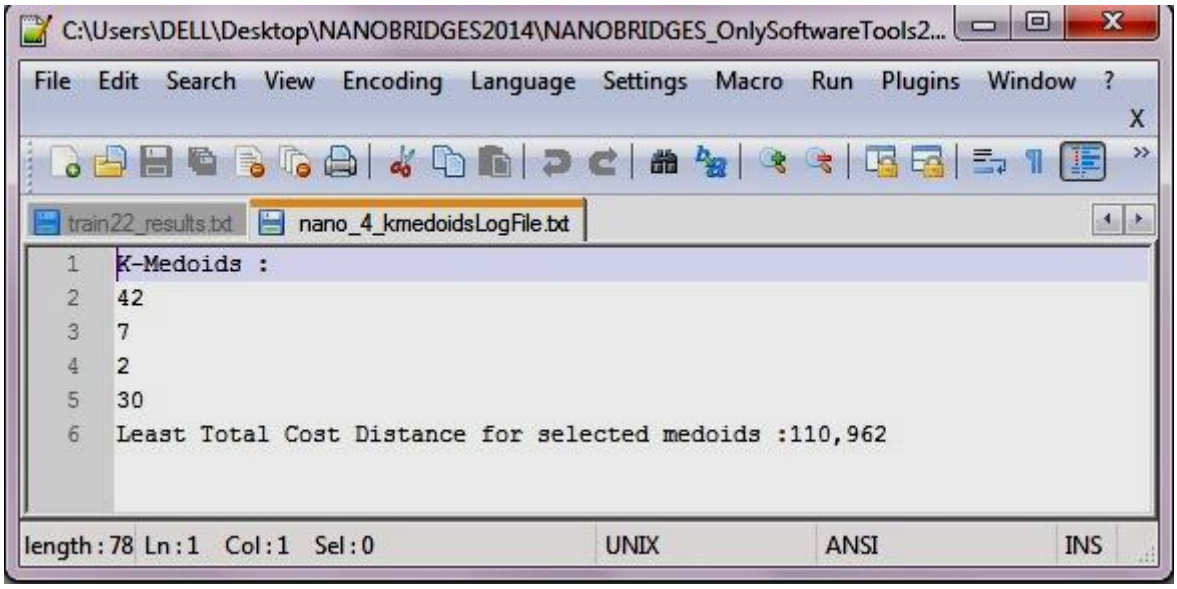
Snapshot 3



DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH	EI
Atype_O	Atype_N	Atype_N	Atype_N	Atype_N	Atype_N	Atype_N	Atype_F	Atype_Br	Atype_S	Fh2o	Foct	LogP	MR	MolRef	Cluster
0	1	0	0	0	0	0	0	0	1	-19,2	-18,6	-0,44	37,601	70,251	3
0	0	1	0	0	0	0	0	0	1	-15,254	-15,54	0,21	39,582	75,744	3
0	0	0	1	0	0	0	0	0	1	-11,307	-12,48	0,86	41,563	79,979	4
0	1	0	0	0	0	0	0	0	1	-19,11	-19,11	0	42,975	75,292	3
0	0	1	0	0	0	0	0	0	1	-15,164	-16,05	0,65	44,956	80,785	3
0	0	0	1	0	0	0	0	0	1	-11,217	-12,99	1,3	46,937	85,02	4
1	1	0	0	0	0	0	0	0	1	-22,856	-22,27	-0,43	46,197	76,714	2
1	0	1	0	0	0	0	0	0	1	-18,91	-19,21	0,22	48,178	82,207	3
1	0	0	1	0	0	0	0	0	1	-14,964	-16,15	0,87	50,159	86,442	4
0	1	0	0	0	0	0	0	0	1	-28,395	-27,44	-0,7	40,328	71,945	3
0	0	1	0	0	0	0	0	0	1	-24,448	-24,38	-0,05	42,309	77,438	3
0	0	0	1	0	0	0	0	0	1	-20,502	-21,32	0,6	44,29	81,673	3
0	1	0	0	0	1	0	0	0	1	-21,018	-19,6	-1,04	35,083	75,988	3
0	0	1	0	0	1	0	0	0	1	-17,072	-16,54	-0,39	37,064	81,482	3
0	0	0	1	0	1	0	0	0	1	-13,125	-13,48	0,26	39,045	85,716	4
0	1	0	0	0	0	0	0	1	1	-21,066	-20,48	-0,43	44,493	77,873	3
0	0	1	0	0	0	0	0	1	1	-17,12	-17,42	0,22	46,474	83,367	3

1. ***Output _.xlsx file (snapshot 3):*** The generated excel sheet (*.xlsx/xls/csv*) will comprise of serial number column and descriptor columns from the input file. One additional column of cluster information (last column) will also be present.
2. ***Log file .txt (snapshot 4):*** This log file consists of information about the selected 'k' medoids and the least total cost distance.

Snapshot 4



Reference:

1. Park, Hae-Sang, and Chi-Hyuck Jun. "A simple and fast algorithm for K-medoids clustering." *Expert Systems with Applications* 36.2 (2009): 3336-3341.

Java External Library Used

Apache POI – the Java API for Microsoft Documents

- Available at http://poi.apache.org/

XMLBeans

- Available at http://xmlbeans.apache.org/

Disclaimer

For academic purpose only.

The program AD-MDI has been developed in Java language and is platform independent. The software is validated on known data sets. Please report for discrepancy of result for any other dataset. Contact us at any of the following addresses:

Dr. Tomasz Puzyn,
NanaBRIDGES Project Coordinator,
Faculty of Chemistry,
University of Gdansk,
Gdansk,
Poland 80-952
Email Id: puzi@qsar.eu.org

Dr. Kunal Roy,
Drug Theoretics and Cheminformatics Lab.,
Dept. of Pharmaceutical Technology,
Jadavpur University,
Kolkata, West Bengal,
INDIA-700032
Email Id: kunalroy_in@yahoo.com

Software Developer details:

Pravin Ambure,
Research Scholar,
Drug Theoretics and Cheminformatics Lab.,
Dept. of Pharmaceutical Technology,
Jadavpur University,
Kolkata, West Bengal,
INDIA-700032
E-mail Id: ambure.pharmait@gmail.com (*for any queries regarding the tool)