DPClusOST: A Software Tool for General Purpose Graph Clustering

Mohammad Bozlul Karim (1), Nobutaka Wakamatsu (1) and Md. Altaf-Ul-Amin (1)

Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-0192 Japan

(Received June 1, 2017; Accepted June 13, 2017)

Modern world is incorporating highly connected heterogeneous data due to information sharing through computer and communication technology. These data lead to a complex relation where drilling down and mining are needed for understanding the actual meaning of data. Today any modern computational technique uses graph clustering as a sophisticated technology for data analysis. In this paper we implement a generalized graph clustering algorithm DPClusO with easy operating procedure and clear visualization techniques. DPClusO is enhanced version of DPClus algorithm where overlapping property of clusters is taken into consideration along with density and periphery tracking. User can select different parameters and visualization attributes to render cluster set, single cluster, hierarchical graph etc. and save these data in image and text formats. This paper discusses step by step operation of the proposed software tool using an example network of metabolites collected from KNAPSAcK database. This tool successfully generated cohesive groups of structurally similar metabolites. The tool can be used for analysis of network data of any field of studies.

Key Words: Cluster, metabolite, density, cluster property, overlapping coefficient

1. Introduction

Network analysis is a comprehensive term used in different fields of studies. Integration of computer aided network analysis tools in different fields such as systems biology, economics, sociology, geography, social psychology, business study helps to get quick findings of divergent similarity patterns, links, relations, even predictions. It also helps to extract kernel information of a complex network dominating the overall interactions.

Expression of macro level attribute into huge pile of smaller pieces of data to better understanding the phenotype of a living organism leads to systems biology. Systems biology deals with huge interaction data of genome, transcriptome, proteome, metabolome and so on. These data construct complex networks like metabolic pathways, gene regulatory networks, protein-protein interaction networks etc. Understanding these complex systems are now becoming possible due to the increasing speed and capacity of modern computers. In systems biology, network analysis is used for function prediction of omics molecules, protein

* amin-m@is.naist.jp
complex detection, interaction detection, disease diagnosis, discovery of new drugs and comparison of different biological mechanisms [1-2]. In economics, network analysis is an increasing field to understand economic phenomena, behaviors or patterns of interaction, financial contagions, socio business network [3,4]. Modern sociology uses the social network analysis (SAN) to analyze friendship, acquaintance kinship and cultural similarity. Sociogram is a graphical representation of such network where nodes represent individuals and links are interpersonal relations [5]. Business network analysis is used in business study as diagnostic methodology that measures the capacity of an organization to effectively engage in its activity. It also uses the social network analysis and organizational network analysis [6]. Different types of network clustering algorithms such as MCL [7], CFinder [8], Affinity clustering [9] along with DPClus [10], DPClusO [11] have emerged. In this work we have developed a GUI tool based on DPClusO algorithm for clustering of simple graphs. The aim of this tool is to implement graph clustering and visualize the binary relational data into two dimensional graphical view of distinct clusters and hierarchical graphs with many useful options. DPClusO is enhanced version of DPClus algorithm which determines a densely connected region of a graph as a cluster and allows overlapping clusters. DPClusO algorithm is robust and memory efficient compared to DPClus hence it takes less time to generate clusters. Throughout the illustration we use a chemical structure similarity based network of metabolites from KNApSAcK database [12]. The generated clusters by our developed system correspond to structurally and functionally similar metabolite groups. However this software tool can be used to analyze big undirected simple graphs in different fields of research and study.

2. DPClusO Algorithm

DPClusO is a simple undirected graph clustering algorithm which can generate overlapping clusters ensuring coverage i.e each node goes to at least one cluster. Here we briefly discuss the DPClusO algorithm. The detail of the algorithm can be found in [11].

2.1. Relevant Definitions. We mention here definitions for some terms from our previously published paper [11] which are also relevant to this work.

Density: If any cluster \( k \) has \( |N_k| \) number of nodes and \( |E_k| \) number of edges then the density \( d_k \) is defined as follows.

\[
d_k = \frac{|E_k|}{|N_k|_{\text{max}}} = \frac{2|E_k|}{|N_k|(|N_k| - 1)}
\]  

Here \( |E_k|_{\text{max}} \) is the maximum possible number of edges in the cluster. Density of a cluster ranges from 0 to 1.

Cluster Property: If any cluster \( k \) has \( |N_k| \) number of nodes then the cluster property \( c_{p_{nk}} \) [3] of any neighboring node \( n \) with respect to cluster \( k \) is defined as follows

\[
c_{p_{nk}} = \frac{|E_{nk}|}{d_k \times |N_k|}
\]  

Here \( |E_{nk}| \) is the total number of edges between the node \( n \) and each of the nodes of cluster \( k \). A neighboring node having higher value of cluster property indicates that it has high priority to be included in the cluster while a lower value indicates that it may be part of the cluster periphery.

Overlapping Coefficient: Overlapping coefficient between two clusters \( a \) and \( b \) is defined by

\[
OV = \frac{|i|^2}{|a| \times |b|}
\]

where \( i \) is number of common nodes between \( a \) and \( b \). \(|a|\) and \(|b|\) are the number of nodes in the clusters.

**Weight of an Edge.** In a graph \( G(N, E) \) the weight \( w_{uv} \) of an edge \( (u, v) \in E \) is the number of the common neighbors of the nodes \( u \) and \( v \) [13].

**Weight of a Node.** In a graph \( G(N, E) \) the weight \( w_n \) of a node \( n \) is the sum of the weights of the edges connected to the node, that is, \( w_n = \Sigma w_{nu} \) for all \( u \) such that \((n, u) \in E\) [13].

2.2. Flowchart of the Algorithm: The input to the DPClusO algorithm is the adjacency list of an undirected simple graph. The flowchart of figure 1 shows the outline of the algorithm. We briefly discuss the important terms of the flowchart in the following.

*Termination Check:* Each iteration of the algorithm removes the edges from main graph whose both nodes belong to the cluster generated in the current iteration. Algorithm terminates when there remains no edge in the main graph.

*Seed Selection:* Initially a seed is selected based on highest node weight from main graph. If weight of every node of remaining graph is zero then seeds are selected based on highest degree. With some exceptions a node that is already part of a cluster is not allowed to be a seed again.

*Priority Neighbor Selection:* The sum of weight of edges between a neighboring node and cluster nodes are calculated. For each neighboring node the highest
weighted node is taken as priority node. If no unique node is found then number of links between a neighboring node and cluster nodes are calculated. If there is a tie also in this stage in priority node selection, then any one is considered as a priority node among the candidates.

Adding a Neighbor to a Cluster: Priority neighbors are added to the cluster on each iteration until the cluster density or cluster property become less than the given threshold values.

2.3 Major Differences with DPclus: DPclusO [11] has been developed for overlapping clustering while DPclus [13-14] was mainly developed for non overlapping clustering with limited capability of overlapping clustering. The DPclusO algorithm guarantees coverage, that is, each node becomes part of at least one cluster, whereas the DPclus algorithm does not ensure coverage. Thus, DPclusO retrieves more information from the network. DPclus works on the adjacency matrix of the network, but DPclusO works on the adjacency list. So DPclusO is faster and memory efficient compared to DPclus.

3. DPclusOST Software

DPclusOST can automatically generate and visualize clusters of an undirected simple network. User can perform filtering after clustering based on overlapping property and cluster size. The main menu of the main window of DPclusOST is shown in figure 2a. Figure 2b shows the sub menus that can be popped up by clicking the main menu items.

3.1 Cluster Generation

3.1.1 Loading Input Network. The simplest way to represent a network is the edge list format and this tool takes the input network represented by this format. Data should be provided by tab separated text file without header information as shown in Fig. 2c. Each line in the

![Flowchart of DPclusO algorithm](image-url)
Jurnal of Computer Aided Chemistry, Vol.18 (2017) 79

Figure 2: Main menu, sub menus and different operational buttons of DPClusOST.
text file represents an edge in term of the pair of nodes it connects. By clicking sub menu item "File Load" from menu Item "Graph Data" (Figure 2b), data load form will appear. User has to click "Load File" button (Figure 2d, "L") to open file browser to select the desired data file. After loading from text file, data will be shown in "Data from file" tabbed pane (Figure 2d "P1"). For illustrating the DPclusO ST, we are using a structural similarity network of metabolites collected from KNAPSACK database consisting of 9129 nodes and 10,000 edges.

3.1.2 Conversion to Adjacency List: Clicking "Convert" button (Figure 2d, "A") will convert the network from edge list format to adjacency list format. Adjacency list data will be shown in "Adjacency List" tabbed pane (Figure 2e "P2").

3.1.3 Clustering: User has to click "Do Cluster" button (Figure 2e, "B") to create the clusters. Clusters will be created and populated in the "Cluster" tabbed pane (Figure 2f, "P3") on descending order of their size. By default a cluster number is given with integer value beginning from one to total cluster number. These numbers are essential for custom filtering of the clusters.

3.2 Cluster Filtering

The numbers of sub graphs of a given density are too many in networks of reasonable size [11]. The algorithm of Georgii et al. [15] attempted to find all possible sub graphs of a given density, but the results indicate that the number of clusters as well as computational time would be very high and would not be feasible for big and dense networks. Therefore DPclusO heuristically try to assign each node to clusters as large as possible and also avoid generating too many nearly identical overlapping clusters to narrow down the candidate clusters. But still the number of clusters generated by DPclusO is relatively large though each of these high-density clusters can be considered as cohesive group of entities. However users may be interested in small number of clusters with limited overlapping and of desired sizes. Therefore, we provide some filtering options with this tool. Filtering can extract a subset of clusters from created cluster set. Big cluster set hinder the visualization of clusters. Filtering options allow user to visualize any portion of created clusters according to following settings.

3.2.1 Overlapping Coefficient: User can enter decimal value within the following range: 0 ≤ ov ≤ 1 in "OV Coff" text field (Figure 2f, "T3") and click "Filter" button (Figure 2f, "C") to reduce the cluster set according to overlapping coefficient. After filtering, no two clusters are more overlapped than the entered OV coefficient. Entering '0' in the "OV Coff" text field will allow no overlapping between any two clusters.

3.2.2 Cluster Size: In the tabbed pane "Cluster", two combo boxes always hold distinct cluster size by descending order. User can enter size range from these two combo boxes to filter clusters of specific sizes.

3.2.3 Cluster Number: User can select any cluster by unchecking the left column of data table and click the cross signed button (Figure 2f, "H") to remove the cluster.

3.2.4 Multiple Cluster Number: Right beside the cross sign button the "M" button (Figure 2f, "G") is used to select multiple clusters. A text box window will appear by clicking the button "M". User can enter desired cluster IDs one by one in a line by ascending order. Once clicking the "Filter" button (Figure 2f, "F") all clusters except the desired clusters (Figure 2g) will be unchecked. User can then click the cross button to remove all undesired clusters.

3.3 Cluster Visualization

Cluster visualization is a technique to represent sub graphs from a network data by using graphical tools. In order to perceive cluster and their interconnectivity visually from big network data this representation is very useful.

3.3.1 Plot Cluster: After the creation of clusters, visualization can be done by clicking "Plot Cluster" button (Figure 2f, "D"). A new GUI window will open on right side with grid layout indicating the cluster boundary. Basically view window is divided by grid layout. For visualization of N clusters, the viewing area is divided into x columns and y rows by solving xy=N and xy =4:3 equations and by rounding or truncating x and y to nearest integers. Thus if total cluster is 48 then grid layout will have 8 columns and 6 rows. If it is 20 then grid layout will have 5 columns and 4 rows. The same ratio is maintained even if the screen size is changed by user. We draw nodes of each cluster with distinct color however all overlapping nodes are represented with red color and each overlapping node is drawn only once. The clusters are arranged row wise according to descending order of size. To aid to vary visualization properties several different options are added via menu items "Screen Size", "Attribute" and "View" (Figure 2b). Below we briefly discuss these options.

(i) Screen Size: Three different types of screen size are included here i.e. default, moderate and large. User can select the screen size by clicking the appropriate size.

(ii) Attribute: This menu item has four different options to make the visualization user friendly so that user can easily interpret each node and edge in a cluster as follows.
a) "Edge" property option set the different color level of edges. With large number of edges between a cluster and its neighbors make the graph a little bit crowded. Making different choice from "deep" to "zero" will give better viewing of the graph.

b) "Font" property allow user to set the font of the node label. By selecting the font name, font size and font type user can set the font property of a node label.

c) "Back Color" property allows user to set the background color of GUI window with two different color types (i.e. white, button face).

d) "Unique Node color" property allows user to set same color for the nodes of a cluster in magnifying window.

(iii) View: This menu item has three different options to place the node position in a cluster region with different coordinate patterns as follows.

a) "Random" pattern set the node coordinates of a cluster in random way using the rectangular height, width of GUI grid window.

Figure 3: Filtering and visualizing. (a) Without any filter, total 3915 clusters. (b) Filtering with OV coff = 0.25 and cluster size >=4, total 537 clusters. (c) Filtering with OV coff = 0.25 and cluster size >=7, total 67 clusters. (d) Filtering with OV coff = 0.25 and cluster size >=10, total 21 clusters.
Figure 4: (a) Cluster generated with density=0.5, cluster property =0.5 and filtered using OV coff = 0.35, size>=12. Total 18 clusters. (b) Magnified view of cluster 11. (c) Extracted overlapping nodes. (d) Circular layout of cluster11.

b) "Circular" pattern set the node coordinates of a cluster using an elliptic equation whose axes fit rectangular GUI window of the cluster with maximum possible size. All node coordinates are fitted to the periphery of the ellipse.

c) "Circular Random" pattern combines the effect of random axis length and an elliptic equation to set coordinates back and forth on the periphery of the ellipse.

3.3.2 Single Cluster Visualization: User can magnify and make better view of a single cluster by double clicking the grid region of the cluster. A new window will open with magnified view of the grid (Figure 4b). It is very important to open single cluster window to view all nodes of a cluster. First double click on the magnified grid will extract all remaining overlapping nodes of the selected cluster from its overlapping neighbors (Figure 4c) and second double click will rearrange the nodes according to different view attributes (Figure 4d).

3.3.3 Hierarchical Graph: User can visualize the hierarchical graph by clicking the "Plot H Graph" button (Figure 2f, "E"). In a hierarchical Graph each cluster is considered as a node. Node radius is calculated by
formula $r = \log(N) \times 10$ where $N$ is total number of nodes in a cluster. The edges between non overlapping nodes of two clusters are considered to determine edge thickness between hierarchical graph nodes. In GUI window, edge line thickness is calculated by formula $d = \log(E) \times 10$ where $E$ is total number of edges between non overlapping nodes of two clusters. Also by selecting different visualization options from menu user can make the GUI window more suitable. In order to arrange the nodes of the hierarchical graph according their size user has to select the "Circular" option from view menu and click the "Plot H Graph" button (Figure 2f, "E"). All nodes of hierarchical Graph will be arranged according to their size in circular fashion (Figure 5b). User can drag and arrange the nodes in view window. An individual node of hierarchical graph can be visualized by double clicking on the node. A new GUI window will appear with actual view of selected node and neighbor cluster nodes (Figure 5c). Red color edges indicate the relation between adjacent clusters and nodes of selected cluster. These relations are edges between all non overlapping nodes. Relation line thickness is kept constant between big blue node and small actual nodes even if the possibility of multiple edges from a small actual node to big blue node. All green nodes indicate non overlapping nodes whereas red nodes are overlapping nodes. Red nodes can be further divided into two groups by selecting two different options from hierarchical graph menu. Selecting "Neighbor Overlapping", the nodes which are overlapped with respect to adjacent neighbor in hierarchical graph will be colored by red (Figure 5d). As hierarchical graph are drawn considering the edges between non overlapping nodes of the clusters of original graph there may be a possibility that two cluster have some overlapping node but no edges between non overlapping nodes. Taken consideration of the fact, "All Overlapping" sub menu item is added to the "H graph" menu (Figure 2b). Selection of this sub menu item will make node color red which are overlapped to any cluster in the graph along with neighbor clusters (Figure 5e).
4. Brief Discussion of Metabolite Clusters

We successfully generated clusters using a structural similarity network of metabolites collected from KNAPSAcK database using density 0.6, cluster property 0.5 and filtered using overlapping coefficient 0.05. Total 2714 clusters were found with minimum size 2 to maximum size 16. After filtering with minimum size 12 we found biggest 6 clusters. Supplementary figure S1 shows metabolites included in top six cluster groups. Cluster 1 contains 16 elements which are all of chemical class "Triterpenoid saponins". Cluster 2 contains 16 elements which are all of chemical class "Proanthocyanidins". Cluster 3 contains 15 elements which are all of chemical class "Acetogenin". Cluster 4 contains 14 elements which are all of chemical class "Triterpene". Cluster 5 contains 14 elements which are all of chemical class "Flavanonol". Cluster 6 contains 12 elements which are all of chemical class "Saponin". These results indicate that DPClusOST successfully clustered structurally similar metabolites, also it can be concluded that structurally similar metabolites belong to specific types and functions which fact is supported by other previous studies [16-18].

5. Output Files

Menu option "Save" provides the facility of saving different information to image files or text files (Figure 2b). Eight different types of output files with four types of image files and four types of text files can be created by user choice. Before saving an image user can rearrange the nodes of a graph by mouse dragging. Also edge color setting, node font setting from menu will give the image a clear and aesthetic view.

5.1 Multiple Cluster: "Multiple Cluster" (Figure 2b) sub menu item is used to create "Multiple_cluster.png" image file of the cluster generated by DPClusOST. User has to activate the GUI window of cluster by clicking over the window and then click "Multiple Cluster" sub menu item to save the image. File save option with default name "Multiple_cluster.png" will appear. User can change the file name also. Once the save button is clicked, graphics from selected window will be saved to the image file.

5.2 Single Cluster: "Single cluster" (Figure 2b) sub menu item allow user to save the image of magnified single cluster window. User can activate the desired window by clicking over it and by selecting the "Single Cluster" sub menu item to save the image. The default filename in this case is "Single_cluster.png".

5.3 Hierarchical Graph: GUI view of hierarchical graph of a given cluster set can be saved in an image file with default name "Hierarchical_Graph.png". User should activate the window of hierarchical graph and select this sub menu item to save the image.

5.4 Hierarchical Node: Detail view of a node from hierarchical graph window can be saved by using this sub menu item. Once user creates the detail view by double clicking the node from hierarchical graph and

---

Figure 6: Different text file from DPClusOST.
activate the window, image can be saved by clicking this sub menu item.

5.5 Adjacency List: This sub menu item is used to save the adjacency list of the graph in a text file. After being created from binary relational input text file the adjacency list populate the tabbed pane labeled by "Adjacency List". By clicking this sub menu item user can save the data with following format: First column indicates individual nodes.

| Cluster No | Cluster Size | Adjacency List |
|------------|--------------|----------------|
| 04         | 16 (208)     | (354, 326, 339, 329, 325, 324, 323, 322, 321, 320, 319, 318, 317, 316, 315, 314, 313, 312, 311, 310, 309, 308, 307, 306, 305, 304, 303, 302, 301, 300, 299, 298, 297, 296, 295, 294, 293, 292, 291, 290, 289, 288, 287, 286, 285, 284, 283, 282, 281, 280, 279, 278, 277, 276, 275, 274, 273, 272, 271, 270, 269, 268, 267, 266, 265, 264, 263, 262, 261, 260, 259, 258, 257, 256, 255, 254, 253, 252, 251, 250, 249, 248, 247, 246, 245, 244, 243, 242, 241, 240, 239, 238, 237, 236, 235, 234, 233, 232, 231, 230, 229, 228, 227, 226, 225, 224, 223, 222, 221, 220, 219, 218, 217, 216, 215, 214, 213, 212, 211, 210, 209, 208, 207, 206, 205, 204, 203, 202, 201, 200, 199, 198, 197, 196, 195, 194, 193, 192, 191, 190, 189, 188, 187, 186, 185, 184, 183, 182, 181, 180, 179, 178, 177, 176, 175, 174, 173, 172, 171, 170, 169, 168, 167, 166, 165, 164, 163, 162, 161, 160, 159, 158, 157, 156, 155, 154, 153, 152, 151, 150, 149, 148, 147, 146, 145, 144, 143, 142, 141, 140, 139, 138, 137, 136, 135, 134, 133, 132, 131, 130, 129, 128, 127, 126, 125, 124, 123, 122, 121, 120, 119, 118, 117, 116, 115, 114, 113, 112, 111, 110, 109, 108, 107, 106, 105, 104, 103, 102, 101, 100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86, 85, 84, 83, 82, 81, 80, 79, 78, 77, 76, 75, 74, 73, 72, 71, 70, 69, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

5.6 Cluster Info: After creating and filtering the cluster, user can save detail cluster information by clicking this sub menu item. A text file will be generated with following format; first column indicates the cluster ID, second column cluster size (total nodes), third column total edge, forth column cluster density and fifth column shows cluster node labels.

5.7 Hierarchical Graph Info: After creating the hierarchical graph by clicking the button "Plot H Graph" this sub menu item will allow saving hierarchical graph information. Default name for this file is "ClusterHierarchyInfo.txt", according to the following format; first column indicates the cluster no, second column the cluster size, third column adjacent cluster list and fourth column shows cluster node list (Figure 6c).

5.8 Node Distribution: An individual node and the clusters to which it belongs are arranged in a text file (Figure 7). User can save this file with default name "NodeDistribution.txt", according to the following format; first column indicates the edge label, second column the frequency of the node occurrence and third column shows the set of cluster IDs.

6. Conclusion

Our main goal was to develop a software tool implementing DPClusO algorithm for graph clustering that can be easily operated. One of the biggest challenges for graph operation using computer technology is to construct a big graph using appropriate memory structure and traverse through it with minimal time. To overcome this problem we utilized the adjacency list structure. This tool is built with flexibility of selecting different parameters like density, cluster property for generating clusters. Clusters can be filtered by overlapping coefficient or size. Different attributes of
GUI display options make the visualization easily understandable along with the saving of graphical and textual information of clusters. The system has been tested with different networks of up to 98,000 nodes. In our present work we demonstrated clustering of a network of metabolites. We verified the structures of the metabolites of biggest 6 clusters using KNApSAcK database and found structural and functional similarity among nodes in individual clusters. We believe that this tool can be applied to analyze networks of many other applications where searching similar pattern, structure or cohesive groups are a requirement. This is the first version of the system. We will update the system according to the user feedback time to time and wish to release latest version incorporating the performance issue or any error exception.

Acknowledgements

This work was supported by the National Bioscience Database Center in Japan; the Ministry of Education, Culture, Sports, Science, and Technology of Japan (16K07223 and 17K00406) and NAIST Big Data Project.

References

[1] M.Altaf-Ul-Amin, Farit Mochamad Afendi, S.Kanaya, Samuel Kuria Kiboi and Shigehiko Kanaya “Systems Biology in the Context of Big Data and Networks” BioMed Research International Volume 2014, Article ID 428570, 11 pages.

[2] M.Altaf-Ul-Amin, Tetsuo Katsuragi, Tetsuo Sato, and Shigehiko Kanaya “A Glimpse to Background and Characteristics of Major Molecular Biological Networks” BioMed Research International Article ID 540297, 2015

[3] Matthew O. Jackson, “The Past and Future of Network Analysis in Economics” https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2520284

[4] Matthew O. Jackson, “The An Overview of Social Networks and Economic Applications” http://web.stanford.edu/~jacksonm/socialnetecon-chapter.pdf

[5] Linton C. Freeman, "THE DEVELOPMENT OF SOCIAL NETWORK ANALYSIS" Empirical Press Vancouver, BC Canada

[6] Graham Durant-Law "Using Business Network Analysis Techniques in Project Management"

[7] Stijn van Dongen. “MCL - a cluster algorithm for graphs” http://micans.org/mcl

[8] Balázs Adámek, Gergely Palla, Illés J. Farkas, Imre Derényi, Tamás Vicsek "CFinder: locating cliques and overlapping modules in biological networks"

[9] Brendan J. Frey; Delbert Dueck (2007). "Clustering by passing messages between data points". Science. 315 (5814): 972–976

[10] Md. Altaf-Ul-Amin, Hisashi Tsuji, Ken Kurokawa, Hiroko Asahi, Yoko Shinbo and Shigehiko Kanaya "DPClus A density-periphery based graph clustering software mainly focused on detection of protein complexes in interaction networks". Information and Communication Technology, 2007. ICICT ’07

[11] M.Altaf-Ul-Amin, M. Wada, S.Kanaya, “Partitioning a PPI Network into Overlapping Modules Constrained by High-Density and Periphery Tracking” International Scholarly Research Network, vol. 2012, Article ID 726429, 2012.

[12] Yukiko Nakamura, Aziza Kawsar Parvin1, Naoaki Ono1, Aki Hirai Morita, Tetsuo Sato1, Tadao Sugiuera, Md. Altaf-Ul-Amin1, Shigehiko Kanaya, Farit Mochamad Afendi, Ken Tanaka, "KNApSAcK Metabolite Activity Database for Retrieving the Relationships Between Metabolites and Biological Activities". Plant Cell Physiol. 2014 Jan;55(1)

[13] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya, “Development and implementation of an algorithm for detection of protein complexes in large interaction networks,” BMC Bioinformatics, vol. 7, article 207, 2006.

[14] M. Altaf-Ul-Amin, H. Tsuji, K. Kurokawa et al., “A density periphery based graph clustering software mainly focused on detection of protein complexes in interaction networks,” Journal of Computer Aided Chemistry, vol. 7, pp. 150–156, 2006.

[15] E. Georgii, S. Dietmann, T. Uno, P. Pagel, and K. Tsuda, “Enumeration of condition-dependent dense modules in protein interaction networks,” Bioinformatics, vol. 25, no. 7, pp. 933–940, 2009.

[16] Azizan Azamini Abdullah, Md. Altaf-Ul-Amin, Naoaki Ono, et al., “Development and Mining of a Volatile Organic Compound Database,” BioMed Research International, vol. 2015, Article ID 139254, 13 pages, 2015. doi:10.1155/2015/139254

[17] Junshui Ma et al. “Deep Neural Nets as a
Method for Quantitative Structure–Activity Relationships, J. Chem. Inf. Model., 2015, 55 (2), pp 263–274 DOI: 10.1021/ci500747n

[18] Sheridan, R.P. et al., 2016. "Extreme Gradient Boosting as a Method for Quantitative Structure-Activity Relationships". Journal of Chemical Information and Modeling

[19] http://clipartpng.com/?1095,green-cartoon-tree-png-clip-art

[20] G. D. Bader and C. W. V. Hogue, “An automated method for finding molecular complexes in large protein interaction networks,” BMC Bioinformatics, vol. 4, article 2, 2003.

Supplementary Materials

Supplementary figure S1. The list of top 6 clusters of KNAPSAcK metabolites generated by DPClusOST (Shown in the next pages).
| Compound ID | Name                  | Molecular formula |
|-------------|-----------------------|-------------------|
| C00029754   | Assamsaponin E        | C59H92O26         |
| C00029753   | Assamsaponin D        | C59H92O27         |
| C00029752   | Assamsaponin C        | C59H90O27         |
| C00032332   | Theasaponin E8 (+)    | C59H90O27         |
| C00032328   | Theasaponin E1        | C59H90O27         |
| C00030275   | Floratheasaponin G    | C60H94O26         |
| C00030269   | Floratheasaponin A    | C59H92O26         |
| C00029751   | Chakasaponin I        | C59H92O26         |
| C00032322   | Theasaponin A2 (+)    | C59H92O27         |
| C00032329   | Theasaponin E2 (+)    | C59H90O27         |
| C00032331   | Theasaponin E4 (+)    | C59H90O27         |
| C00032333   | Theasaponin E7 (+)    | C59H90O27         |
| C00030277   | Floratheasaponin (+)  | C60H94O26         |
| C00032334   | Theasaponin E2 (+)    | C59H90O27         |

Figure S1(a): Cluster 1, 16 Elements, Triterpenoid saponins
Figure S1(b): Cluster 2, 16 Elements, Proanthocyanidins or Condensed tannins
| Compound          | Molecular Formula |
|-------------------|-------------------|
| Squamocin F       | C37H66O7          |
| Squamocin         | C37H66O7          |
| Asiminin          | C37H66O7          |
| Asiminocin (+)    | C37H66O7          |
| Asiminocin        | C37H66O7          |
| Asimicin          | C37H66O7          |
| Asitribin         | C37H66O7          |
| Bullatin          | C37H66O7          |
| Bullatin          | C37H66O7          |
| Bullatacin (+)    | C37H66O7          |
| Bullatacin        | C37H66O7          |
| Guanacone         | C37H64O7          |
| Guanacone         | C37H64O7          |
| Rolliniastatin 1  | C37H66O7          |
| Squamocin D       | C37H66O7          |
| Trilobacin        | C37H66O7          |

**Figure S1(c): Cluster 3, 15 Elements, Acetogenin**
| Compound Code | Name                                      | Molecular Formula |
|---------------|-------------------------------------------|-------------------|
| C00033122     | Loranthol                                 | C30H50O2          |
| C00041268     | 3alpha,20-Lupandiol (-)-3alpha,20-Lupandiol | C30H52O2          |
| C00023789     | Glochidonol                               | C30H48O2          |
| C00034033     | Lup-20(29)-ene-3beta,16beta-diol (+)-4,up-20(29)-ene-3beta,16beta-diol | C30H50O2          |
| C00037440     | Lup-20(29)-ene-1beta,3beta-diol           | C30H50O2          |
| C00034197     | Resinone                                  | C30H48O2          |
| C00003740     | Betulin                                   | C30H50O2          |
| C00042563     | Glochidiol                                | C30H50O2          |
| C00021224     | Betulousaldehyde Betunol (+)-Betunol      | C30H48O2          |
| C00042123     | 3-epi-Betulin (+)-3-epi-Betulin           | C30H50O2          |
| C00042124     | 3-epi-Betulinic aldehyde                 | C30H48O2          |
| C00042283     | Betulone (+)-Betulone                    | C30H48O2          |
| C00034871     | Lup-20(29)-ene-3beta,30-diol (-)-Lup-20(29)-ene-3beta,30-diol | C30H50O2          |
| C00041266     | 30-Hydroxylup-20(29)-en-3-one (+)-30-Hydroxy lup-20(29)-en-3-one | C30H48O2          |

*Figure S1(d): Cluster 4, 14 Elements, Triterpene*
Figure S1(e): Cluster 5, 14 Elements, Flavanonol
Figure S1(f): Cluster 6, 12 Elements, Saponin