Supplementary file: Multi-factored gene-gene proximity measures exploiting biological knowledge extracted from Gene Ontology : application in gene clustering

Sudipta Acharya[1], Sriparna Saha[1], Prasanna Pradhan[2]
[1]Department of Computer Science and Engineering, Indian Institute of Technology Patna, India
[2]Department of Computer Applications, Sikkim Manipal Institute of Technology
Corresponding author: S. Acharya (email:sudiptaacharya.2012@gmail.com)

I. Chosen clustering algorithms

A. AMOSA based clustering algorithm

Here the problem of clustering a given gene data set is addressed by simultaneously optimizing a set of cluster validity indices capturing different cluster qualities. In order to make this simultaneous optimization feasible, and also to identify a set of trade-off solutions, the search capability of a popular multi-objective optimization or MOO technique, AMOSA (Archived Multi Objective Simulated Annealing) [3] is accomplished. It has been shown in the literature that AMOSA excels in the field of MOO as compared to several other existing multi-objective evolutionary algorithms. The steps of AMOSA based proposed clustering technique are mentioned below,

1) String Representation and Archive Initialization

AMOSA [3] utilizes the concept of string to represent a particular solution. At the beginning of execution it initializes the archive with some random solutions. Each archive member represents one complete clustering solution. The length of the archive member varies over a range. Suppose our chosen data set contains n number of genes and each gene has d number of total GO terms (for biological process, cellular component, molecular function). n and d are specific to a data set.

Let us assume that archive member i represent the centroids of K_i clusters. Then the array or archive member has length l_i where $l_i = d * K_i$.

Each data point represents a gene of d number of total GO terms and each cluster centroid c_k is defined by a vector of d GO terms.

Each centroid used in string encoding is atomic in nature i.e., during mutation if we insert one centroid then all the contained annotation values will be inserted. Similarly if we perform deletion during mutation, all annotation values of the chosen centroid will be deleted.

The number of centroids, K_i, encoded in a string i is chosen randomly between two limits K_{min} and K_{max}. The following equation is utilized to determine this value:

$$K_i = (rand() \text{mod}(K_{max} - 1)) + K_{min} \tag{1}$$

Here, $rand()$ is a function returning a random integer number and K_{max} is the upper-limit of the number of clusters. The minimum number of clusters (K_{min}) is assumed to be 2. The number of whole clusters present in a particular string/ member of archive can therefore vary in the range of 2 to K_{max}. For the initialization purpose, these K_i cluster centroids represented in a string are some randomly generated genes from the cancer data set.

2) Assignment of Points and Computation of Objective Functions

After the initialization of archive members with some randomly generated cluster centroids, assignment of n genes or data points (where n = total number of genes in a particular data set) to different clusters is performed. Next, we compute two cluster quality measures, XB index[14], PBM index[2], which are used as two objective functions for each solution or string. Thereafter using the search methodology of AMOSA we simultaneously optimize these two objective functions.

1) **Membership of Genes to Different Clusters:**
In this part, the assignment is done based on any one of our proposed and existing semantic similarity or distance measures (as mention in Section II and IV). For a particular dataset the gene-gene similarity/ distance measure can be found from generated corresponding similarity matrices as described in Section V.A.3). A maximum similarity value between a point (here gene) and a cluster center results in assigning that point to the given cluster. Please note that here the cluster center is one of the points in the dataset itself.
2) **Objective Functions:**
To measure the quality of each solution, two objective functions are calculated. Both are functions of cluster compactness and separation which are shown in Figurecompact. To get some optimized solutions, value of XB index corresponding to a particular solution should be minimized and the value of PBM index should be maximized. These two objective functions are optimized simultaneously using the search capability of AMOSA. The mathematical formulations of these objective functions are given below,
   - **XB index:** Xie and Beni[14] proposed a cluster validity index (XB) which is a function of compactness and separation. Low value of compactness and high value of separation indicate good quality, well separated clusters. Hence, the most desirable partitioning is obtained by minimizing the XB index for k=2....K_{max}.

$$XB = \frac{\sum_{i=1}^{K_{max}} \sum_{j=1}^{n} \mu_{ij}^2 \|\overline{x}_j - \overline{c}_i\|^2}{n(min_{i \neq k} \|\overline{c}_i - \overline{c}_k\|^2)} \qquad (2)$$

where, K_{max} = total number of clusters in a solution.
n = total number of data points to be clustered.
μ_{ij}= membership of data point i with respect to cluster j (computed as described in Section I-A2).
x_j = j^{th} data point.
c_i = i^{th} cluster.

- **PBM index:** The PBM index (acronym constituted of the initials of the names of its authors, Pakhira, Bandyopadhyay and Maulik)[2] is calculated using the distances between the points and their centroids and the distances among the centroids themselves. It is an Euclidean distance based cluster validity index which should be maximized for obtaining the correct number of clusters. It is defined as given below:

$$PBM(K) = (\frac{1}{K} \times \frac{\mathcal{E}_1}{\mathcal{E}_{\mathcal{K}}} \times D_K)^p \qquad (3)$$

where K = number of clusters
Here $\mathcal{E}_{\mathcal{K}} = \sum_{k=1}^{K} \sum_{j=1}^{n_k} d_e(\overline{c}_k, \overline{x}_j^k)$
and $D_K = \max_{i,j=1}^{K} d_e(\overline{c}_i, \overline{c}_j)$
where $\overline{c}_j$ = Center of the j^{th} cluster,
$\overline{x}_j^k$ = j^{th} point of the k^{th} cluster.
n_k = total number of data points of the k^{th} cluster.
The power p is used to control the contrast between different cluster configurations. In this article we have kept $p = 2$.

Difference between these two objective functions is that these two capture different aspects of cluster compactness and separation.

3) Search Operators

Performing perturbation operation on clustering solutions helps to explore the search space properly. Therefore, we have used three different mutation operations which are mentioned as follows:

A clustering solution can be changed in three different ways,

- One randomly chosen cluster center encoded in the solution can be replaced by another data point. Among all data points of a dataset one data point (here point means gene) with respect to which the average similarity of all the members of the selected cluster is maximum is nominated to replace the existing center of that cluster.
- A number of encoded clusters in a solution can be decreased by one. This is done by deleting a randomly selected cluster center from the given solution.
- A number of encoded clusters in a solution can be increased by one. This is done by randomly selecting a point from the dataset as the new cluster center and then inserting this in the solution.

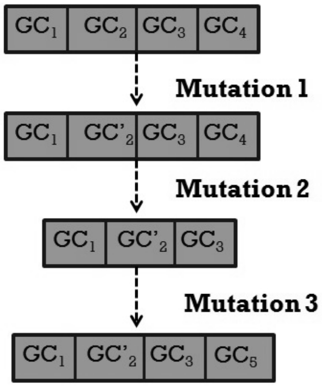


Fig. 1: Three types of mutation operations applied on a string

1) **Mutation 1:** This is used to replace cluster center by a new center. Suppose center of cluster C_i is selected for this mutation operation. Then the new cluster center of cluster C_i is chosen according to any of the following equations.

$$j = argmin_j \frac{\sum_{\forall G_j \notin C_i} \sum_{k=1 \dots p} d(G_j, C_{i_{G_k}})}{|C_i|} \qquad (4)$$

$$j = argmax_j \frac{\sum_{\forall G_j \notin C_i} \sum_{k=1 \dots p} S(G_j, C_{i_{G_k}})}{|C_i|} \qquad (5)$$

Where, $C_{i_{G_k}}$ represents k^{th} gene, G_k, of cluster C_i. p is the number of data points in cluster C_i. $d(G_j, C_{i_{G_k}})$ is the distance between gene G_j and $C_{i_{G_k}}$ according to any of our chosen distance measures in this article. $S(G_j, C_{i_{G_k}})$ is the similarity between gene G_j and $C_{i_{G_k}}$ according to any of our chosen similarity measures in this article.
According to Equations 4 and 5, the center of cluster C_i gets replaced by gene G_j.

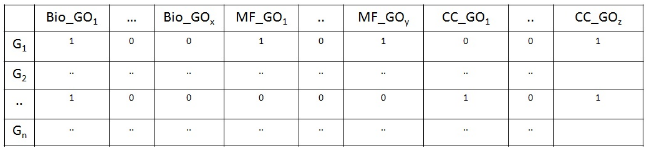
2) **Mutation 2:** In this type of mutation the size of the string is decreased by one. From the string a cluster center is chosen randomly and then deleted. As each cluster center is considered to be indivisible, so by deleting a cluster center all of its dimensional values are removed.

3) **Mutation 3:** This mutation is used to increase the size of the string by one. This is performed by inserting a new center in the string. Similar to 2^{nd} type of mutation here also each center is considered to be indivisible.

In order to illustrate the process further, in Figure 1 we have shown the effect of mutation operations on a particular string. Here GC_i is the center of i^{th} cluster of a particular string. The final length of the string varies with the type of mutation operation applied.

4) Selecting Best Clustering Solution from the Pareto Optimal Front

A set of non dominated solutions is produced by any MOO technique [1] on its Pareto optimal front. Each of the points on the final Pareto optimal front represents one complete clustering solution. Each of these non-dominated solutions corresponds to a complete assignment of all datapoints of chosen data set to different clusters. In the absence of additional information, any of those solutions can be selected as the optimal solution. In this approach we have selected



	Bio_GO$_1$	...	Bio_GO$_x$	MF_GO$_1$	..	MF_GO$_y$	CC_GO$_1$	..	CC_GO$_z$
G_1	1	0	0	1	0	1	0	0	1
G_2	..	..	..	..	..	..	..	..	
..	1	0	0	0	0	0	1	0	1
G_n	..	..	..	..	..	..	..	..	

Fig. 3: Binary *gene-GO term* annotation matrix representation

the best solution using one internal cluster validity index, i.e., Silhouette index [13]. The solution having highest Silhouette index value is selected as the best solution.

B. NSGA-II based clustering

Non-dominated Sorting Genetic Algorithm-II or NSGA-II [6] is a popular multi-objective evolutionary algorithm. Existing literature also suggests that NSGA-II can perform well while optimizing up to four objective functions simultaneously. Motivated by this, in [7] authors have developed a NSGA-II based clustering framework. We have utilized the same framework with the set of objective functions (XB index and PBM index) as used in AMOSA based clustering to automatically cluster the given gene data set using various gene-gene similarity matrices.

C. K-means clustering algorithm

K means clustering algorithm was developed by J. McQueen and then by J. A. Hartigan and M. A. Wong around 1979 [9]. It is one of the simplest unsupervised learning algorithms that solves the well known clustering problem [10], [12]. The procedure follows a simple and easy way to partition a given data set into a certain number of clusters (assume 'K' clusters) fixed a priori. The value of 'K' is chosen according to Section VI.A.3 and Table I of main manuscript. The steps of K-means are mentioned below in brief,

1) Select 'K' points randomly from the set of available points. These points represent initial group centroids.
2) Assign each object to the closest centroid using some distance measure.
3) When all objects have been assigned, recalculate the positions of the 'K' centroids.
4) Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

D. K-medoids clustering algorithm

The K-medoid algorithm is a clustering algorithm related to the K-means algorithm and the medoid shift algorithm. Both K-means and K-medoid algorithms are partitional (breaking the dataset into several groups) in nature. K-means attempts to minimize the total squared error, while K-medoids minimizes the sum of dissimilarities between points which are in a single cluster with respect to the medoid, a point designated as the center of that cluster. In contrast to the K-means algorithm, K-medoids chooses any real data point from the existing cluster as the center. In K-medoids, 'K' is chosen apriori as chosen in K-means algorithm. It is more robust to noise and outliers as compared to K-means because it minimizes a sum of general pairwise dissimilarities instead of a sum of squared Euclidean distances. The most common realisation of K-medoid clustering is the Partitioning Around Medoids (PAM) algorithm and the corresponding steps are summarized as follows:

1) Initialize: randomly select 'K' of the n data points as the medoids.
2) Associate each data point to the closest medoid ("closest" here is defined using any valid similarity/distance measure, in this work we have used seven different similarity and distance measures).
3) For each medoid m:
   For each non-medoid data point o:
   Swap m and o and compute the total cost of the configuration (sum of distances of points to their medoids).
4) Select the configuration with the lowest cost.
5) Repeat steps 2 to 4 until there is no change in the medoid.

E. Hierarchical clustering algorithm

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering (defined by S.C. Johnson in 1967) [11] is as follows:

1) Start by assigning each item to a cluster, so that initially we have N number of clusters where N is the total number of data points. The initial similarity matrix between clusters is same as the initial similarity matrix of the data points.
2) Find the closest (most similar) pair of clusters and merge them into a single cluster, so that the number of clusters is reduced by one.
3) Update the distance matrix between clusters after computing distances (similarities) between the new cluster and each of the old clusters.
4) Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Step 3 can be done in different ways, based on which there are three types of hiearchical clustering techniques: single-linkage, complete-linkage and average-linkage clustering. In single-linkage clustering (also called the connectedness or minimum method), the distance between two clusters is kept equal to the shortest distance between any two points belonging to two different clusters. If similarities between clusters are given as input instead of distances, the similarity between two clusters is considered as the highest similarity between two points belonging to two different clusters. In complete-linkage clustering (also called the diameter or maximum method), the distance between two clusters is kept equal to the largest distance between two points belonging to two different clusters. In average-linkage clustering, the distance between two clusters is kept equal to the average distance between all pairs of points belonging to two different clusters. A variation of average-link clustering is the UCLUS method of R. D'Andrade (1978) [4] which uses the median distance.

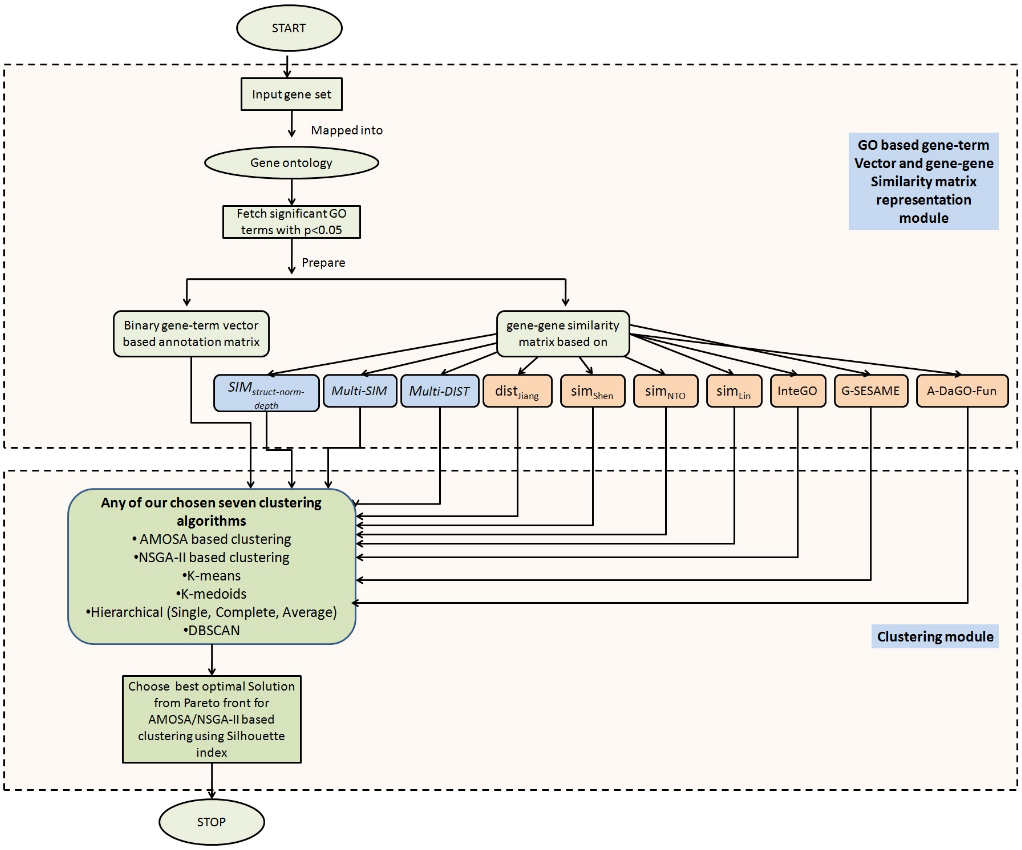

START
Input gene set
Mapped into
Gene ontology
Fetch significant GO terms with p<0.05
Prepare
GO based gene-term Vector and gene-gene Similarity matrix representation module
Binary gene-term vector based annotation matrix
gene-gene similarity matrix based on
SIM struct-norm-depth
Multi-SIM
Multi-DIST
dist Jiang
sim Shen
sim NTO
sim Lin
InteGO
G-SESAME
A-DaGO-Fun
Any of our chosen seven clustering algorithms
• AMOSA based clustering
•NSGA-II based clustering
•K-means
•K-medoids
•Hierarchical (Single, Complete, Average)
•DBSCAN
Clustering module
Choose best optimal Solution from Pareto front for AMOSA/NSGA-II based clustering using Silhouette index
STOP

Fig. 2: Flowchart of the proposed framework

This helps to detect the outliers easily compared to the average distance.

F. DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jrg Sander and Xiaowei Xu in 1996 [8]. It utilizes the concepts of density to identify regions of higher densities separated by regions of low densities. The core idea of this algorithm is to identify three sets of points : outliers (which are not part of any cluster), core points (which are the major components of a particular cluster), border points (which are not the core points but density connected with the core points). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. Another advantage of it is the automatic determination of the number of clusters. But the algorithm heavily suffers from the problem of improper selection of its parameters which can lead to wrong results.

II. Chosen cluster validity measures

In order to evaluate goodness of clustering solutions obtained by several clustering techniques, some external or internal cluster validity measures can be used. As true partitioning information of our chosen dataset is not available so we have utilized two internal cluster validity measures, namely Silhouette index [13] and DB index [5], to quantify the goodness of the partitionings. These are described as follows:

A. Silhouette index

Silhouette index [13] is an internal cluster validity index which we have used in this article to measure the goodness of clustering solutions. It is a function of cluster compactness and separation. Suppose,
x= average distance of a particular point from other points of the same cluster in which that point resides.
y= minimum of average distance of that point to other points of other clusters.
Then the Silhouette width, Sil, can be defined as,

$$Sil = \frac{(x-y)}{max(x,y)} \tag{6}$$

Silhouette index is computed as average of Silhouette width of all data points of a given data set. It can vary from -1 to 1 and a good clustering solution possesses higher Silhouette index value.

B. DB index

DaviesBouldin index or DB index [5] is an internal cluster validity index to measure the goodness of the formed clusters. It is expressed as,
(Summation of within-cluster separation / the between-cluster separation)
Separation within the i^{th} cluster, S_i, is calculated as S_i =

$[\sum_{x\in C_i} d_e(x,c_i)]/n_i$ where n_i denotes the number of points present in cluster C_i, and $d_e(x,c_i)$ is the Euclidean distance between the point x and the center c_i of the i^{th} cluster, C_i. The distance between clusters C_i and C_j , denoted by d_{ij}, is defined as $d_{ij} = d_e(c_i, c_j)$. Then, DB index is defined as,

$$DB = \frac{\sum_{i=1}^{K} R_i}{K} \tag{7}$$

here, $R_i = max_{j,j\neq i}(S_i + S_j)/d_{ij}$. If the obtained clusters are compact in shape and also well separated from each other then the DB index value would be the lowest. A good clustering solution would minimize the DB index value as much as possible.

GO term	Module %	Genome %
transmembrane transport GO:0055085	47.49%	27.32%
cellular response to DNA damage stimulus GO:0006974	53.11%	35.30%
response to chemical GO:0042221	57.73%	38.39%
ion transport GO:0006811	49.73%	36.39%
mitotic cell cycle GO:0000278	58.05%	35.77%
rRNA processing GO:0006364	57.28%	25.44%
carbohydrate metabolic process GO:0005975	46.90%	34.90%

TABLE III: Significant GO terms shared by genes of cluster 2 for *Yeast* dataset produced by AMOSA- clustering

GO term	Module %	Genome %
transcription from RNA polymerase II promoter GO:0006366	49.71%	28.16%
cellular response to DNA damage stimulus GO:0006974	47.55%	25.30%
chromatin organization GO:0006325	46.83%	25.35%
organelle fission GO:0048285	56.83%	24.62%
DNA repair GO:0006281	56.83%	24.58%
regulation of organelle organization GO:0033043	46.47%	24.89%

TABLE IV: Significant GO terms shared by genes of cluster 3 for *Yeast* dataset produced by AMOSA- clustering

GO term	Module %	Genome %
transmembrane transport GO:0055085	58.92%	37.32%
carbohydrate metabolic process GO:0005975	47.38%	34.90%
organelle fission GO:0048285	57.38%	34.62%
cytoskeleton organization GO:0007010	46.46%	23.94%
cell wall organization or biogenesis GO:0071554	55.54%	24.65%

TABLE V: Significant GO terms shared by genes of cluster 4 for *Yeast* dataset produced by AMOSA- clustering

GO term	Module %	Genome %
cellular response to DNA damage stimulus GO:0006974	46.22%	35.30%
cellular amino acid metabolic process GO:0006520	45.85%	33.95%
organelle fission GO:0048285	45.67%	34.62%
protein phosphorylation GO:0006468	24.75%	13.63%
peptidyl-amino acid modification GO:0018193	24.39%	23.41%

TABLE VI: Significant GO terms shared by genes of cluster 5 for *Yeast* dataset produced by AMOSA- clustering

GO term	Module %	Genome %
generation of precursor metabolites and energy GO:0006091	43.50%	23.02%
monocarboxylic acid metabolic process GO:0032787	43.50%	32.96%
sporulation GO:0043934	43.72%	22.73%
chromosome segregation GO:0007059	34.60%	23.46%
cofactor metabolic process GO:0051186	34.81%	13.35%

TABLE VII: Significant GO terms shared by genes of cluster 6 for *Yeast* dataset produced by AMOSA- clustering

REFERENCES

[1] F. Angiulli and C. Pizzuti, "Gene expression biclustering using random walk strategies," in *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 2005, pp. 509–519.

[2] S. Bandyopadhyay and S. Saha, *Unsupervised classification: similarity measures, classical and metaheuristic approaches, and applications*. Springer Science & Business Media, 2012.

[3] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: Amosa," *IEEE transactions on evolutionary computation*, vol. 12, no. 3, pp. 269–283, 2008.

[4] R. G. D'Andrade, "U-statistic hierarchical clustering," *Psychometrika*, vol. 43, no. 1, pp. 59–67, 1978.

[5] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[7] P. Dutta and S. Saha, "Fusion of expression values and protein interaction information using multi-objective optimization for improving gene clustering," *Computers in biology and medicine*, vol. 89, pp. 31–43, 2017.

[8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[9] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[10] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[11] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[12] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[13] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

p values							
Proximity measures	NSGA-II clustering	K-means	K-medoids	Hier-single	Hier-average	Hier-complete	DBSCAN
$dist_{Jiang}$	2.321E-172	2.273E-185	2.563E-231	2.573E-236	2.178E-315	2.754E-126	2.432E-214
sim_{Shen}	1.732E-243	2.1596E-285	2.455E-083	1.464E-237	2.167E-094	2.464E-218	2.273E-194
sim_{NTO}	1.253E-163	2.187E-298	1.678E-247	1.574E-216	2.579E-186	2.679E-245	2.636E-217
sim_{Lin}	2.412E-252	1.234E-291	3.123E-092	2.184E-021	2.54E-341	1.984E-192	1.578E-316
$InteGO$	2.342E-156	3.166E-296	2.734E-172	2.583E-177	1.463E-426	2.562E-251	2.174E-287
$G-SESAME$	1.735E-261	2.183E-291	1.654E-235	2.182E-213	2.183E-73	1.732E-077	2.134E-148
$A-DaGO-Fun$	2.165E-183	2.856E-193	2.674E-231	1.738E-291	2.272E-193	2.73E-291	2.465E-201
$SIM_{norm-struct_{depth}}$	2.173E-313	3.824E-382	3.213E-173	2.184E-328	3.876E-217	2.765E-182	2.684E-187
Multi-SIM	2.173E-292	2.363E-281	2.876E-195	2.653E-194	1.753E-052	3.15E-127	1.323E-183
Multi-DIST	2.153E-291	2.184E-371	2.543E-321	1.854E-218	2.753E-219	2.753E-092	2.262E-166

TABLE I: The p-values produced by t-test comparing DB index by AMOSA based clustering algorithm with other algorithms for all the ten similarity/distance measures for *Yeast* dataset

p values								
Clustering algorithms	$dist_{Jiang}$	sim_{Shen}	sim_{NTO}	sim_{Lin}	$InteGO$	$G-SESAME$	$A-DaGO-Fun$	$SIM_{norm-struct_{depth}}$
AMOSA clustering	2.546E-036	3.564E-173	3.218E-234	2.163E-435	2.334E-271	2.252E-183	2.124E-281	2.845E-382
NSGA-II clustering	1.366E-023	1.27E-092	1.382E-120	1.834E-092	1.284E-212	1.835E-095	2.398E-193	2.162E-183
K-means	2.136E-254	1.65E-193	2.502E-193	1.982E-218	2.184E-129	2.193E-216	1.367E-193	2.549E-367
K-medoids	2.173E-293	3.283E-184	3.273E-193	2.74E-193	2.935E-183	3.262E-269	1.856E-291	2.146E-292
Hier-single	1.546E-187	3.175E-379	2.764E-285	2.754E-317	2.368E-294	3.643E-295	1.272E-294	2.145E-183
Hier-Average	2.595E-196	2.785E-318	1.754E-219	2.368E-296	2.837E-193	2.183E-149	2.184E-295	2.173E-378
Hier-Complete	2.153E-272	3.183E-291	3.173E-347	3.175E-285	3.273E-193	2.193E-295	1.856E-292	2.173E-268
DBSCAN	3.173E-183	2.183E-083	1.366E-193	2.163E-193	2.181E-069	1.377E-282	2.471E-275	2.272E-173

TABLE II: The p-values produced by t-test comparing DB index by *Multi-SIM* measure with other measures for all of the eight chosen clustering algorithms for *Yeast* dataset

[14] X. L. Xie and G. Beni, “A validity measure for fuzzy clustering,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 13, no. 8, pp. 841–847, 1991.