International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

BiETech : Bicluster Ensemble Techniques

Geeta Aggarwal[1], Seema Aggarwal[2]*, Manisha Bansal[3], Neelima Gupta[4]
[1]geeta.gupta@pgdav.du.ac.in, https://orcid.org/0000-0003-4399-9937, PGDAV College, University of Delhi
[2*]Seema.aggarwal@mirandahouse.ac.in, https://orcid.org/0000-0003-3107-0629, Miranda House, University of Delhi
[3]mbansal@ip.du.ac.in, Indraprastha College for women, University of Delhi
[4]ngupta@cs.du.ac.in, https://orcid.org/0000-0002-0870-0824, Department of Computer Science, University of Delhi

***Corresponding Author:** Seema Aggarwal
*Seema.aggarwal@mirandahouse.ac.in, https://orcid.org/0000-0003-3107-0629, Miranda House, University of Delhi

Abstract: Various biclustering algorithms have emerged now a days that try to deliver good biclusters from gene expression data which satisfy a particular objective function. Users are lost in finding the best out of these algorithms. Ensemble techniques come to rescue of these users by aggregating all the solutions and providing a single solution which is more robust and stable than its constituent solutions. In this paper, we present two different ensemble techniques for biclustering solutions. We have used classifiers in one approach and the other approach usesthe concept of metaclustering for forming the consensus. Experiments in this research are performed on synthetic and real gene expression datasets as biologists are interested in finding meaningful patterns in expression of genes. The experiments show that both the approaches proposed in the paper show improvement over the input solutions as well as the existing bicluster ensemble techniques.

Keywords: Bicluster Ensemble; Classifiers; Discriminant Analysis; Support Vector Machine; Mutual Information; Gene Expression Data

1. Introduction

Gene expression data [1] [2] [3] [4] [5] [6] is being studied by a lot of researchers who are interested in finding meaningful patterns in the expression of genes. It has been observed that genes that are responsible for one biological activity behave similarly under a specific subset of samples or conditions. The whole set of conditions is not required for such an activity. Traditional clustering algorithms that identify clusters based on the complete set of conditions are not suitable for extracting such patterns. Biclustering is the term coined by Hartigan [7] and was first used by Cheng and Church [1] for the gene expression data. It refers to simultaneous clustering of genes and samples. Biclusters may overlap on genes, on samples or both as a gene/condition may be responsible for more than one biological activity and hence may be a member of more than one bicluster in the expression data.

Various approaches exist in literature for identifying biclusters in the data. Most of them lack robustness and stability with respect to the random initialization. Tools like BiCAT [8] and BiDeal [9] exists that help users to generate input schemes using different biclustering algorithms by embedding all of them on a single platform. All these algorithms lead to different solutions resulting from different objective functions they optimize. The end user often finds it difficult to select an algorithm that suits best for his/her application. Ensembling is a technique that collects several solutions and generates a consensus that best approximates the input solutions. Ensemble techniques have been successfully used in classification and clustering but ensembling biclusters is relatively a new field and more challenging. The main challenge stems from the fact that the different biclusters that need to be combined involve different sets of conditions.

Two main steps of ensembling are: generation of input solutions and combining the input solutions to generate a consensus function. Two approaches are largely used to design consensus functions in clustering [10] [11] [12] [13] [14] [15] one that establishes label correspondence between the various partitions and then uses a consensus function; second that eliminates the need of label correspondence and computes the consensus function directly. In [16] Hanczar and Nadif (HN) proposed the use of bagging (bootstrapped aggregation) to improve the performance of biclustering schemes. In our previous work [17] [18], we proposed an approach (BiETopti) that uses optimization techniques to generate the consensus. As much of the information is lost in bootstrapping, we ensembled schemes generated without bootstrapping in BiETopti. It was shown that BiETopti outperforms HN both in terms of time and quality. In this paper we propose two approaches; one that tries to improve upon the quality of the biclusters by using classifiers, second one that does away with the need of establishing label correspondence and uses the concept of meta-clusters to form the consensus directly resulting in large improvement in computational time. The proposed approaches outperform BiETopti and hence HN in terms of time as well as quality. However, there is a tradeoff amongst themselves in time and quality. The first approach referred as BiETclassi gives superior quality biclusters than the second approach called BiETmetaclus. However, BiETclassi takes more time than BiETmetaclus as it involves expensive steps like label correspondence and classification techniques like discriminant analysis [19] and support vector machines [20].

The challenge with using classifiers in BiETclassi to predict the labels of gene for the biclustering problem is threefold. One, the biclusters are non-disjoint. Secondly the biclusters are non-exhaustive. Thirdly, the set of attributes (samples) is different for different biclusters. We combine the approaches used for multiclass and multilabel classifiers to address these challenges. BiETmetaclus uses Mutual Information (MI) [21] to find similarity between the biclusters. Though MI has been used as a similarity measure in traditional clustering [22], it has been used for the first time to capture similarity of biclusters.

Experiments were performed on the benchmark datasets of Prelic et al. [4] and on the real expression data of Saccharomyces Cerevisiae (Yeast), Arabidopsis Thaliana, Human Diffuse Large B-cell Lymphoma (DLBCL), and Human Breast Cancer. Saccharomyces Cerevisiae is an ideal dataset for scientists studying lower eukaryotic organisms whereas Arabidopsis Thaliana is the best fit for the ones interested in plant research. To assessthe performance of our algorithm on higher organisms like homosapiens datasets Human Diffuse Large B-cell Lymphoma (DLBCL), and Human Breast Cancer were used. The validation techniques like Biclustering Error (BCE) and Average Rand Index (AvRI) wereused to adjudge the statistical significance of the biclusters obtained in the synthetic data sets whereas GO terms and motif analysis were used to compare the performance on the real datasets in the absence of ground truth.

Remaining paper is organized as follows: the problem is defined in section 2. Section 3discusses the related work. The first approach BiETclassi is presented in section 4. The experimental results of BiETclassi and its comparison with BiETopti are presented in section 5. BiETmetaclus, the second approach is described in section 6. The experimental results of BiETmetaclus and its comparison with BiETopti and BiETclassi are presented insection 7. The two approaches are compared with the existing biclustering algorithms in section 8. Section 9 compares the time complexity of the two approaches with BIETopti. The paper is concluded with the future work in section 10.

2. Problem Definition

Let G be a set of N genes and C be a set of d samples/conditions. Expression matrix E is $N * d$, *where* each row represents the expression of a gene under d samples. For i= 1 to H, H being the number of schemes to be ensembled, let π_i denote the i^{th} biclusteringscheme obtained when E is subjected to a biclustering algorithm. Let k_i denote the number of biclusters in π_i. Thus, $\pi_i = (BC_1, BC_2, ..., BC_{ki})$, where BC_j is a tuple (G_j, C_j), G_j being a subset of genes and C_j a subset of conditions. Note that different biclustering schemes may contain different number of biclusters.

Further $\lambda : E\,(G * C) \rightarrow 2^{\{0...k\}}$ a function that yields a set of labels for each gene condition pair (g_l, c_r). Note that since the biclusters may overlap both on genes and conditions, a (gene, condition) pair may be assigned more than one label. Also, there may be a (gene, condition) pair which does not belong to any bicluster, such a pair is assigned a special label 0. Let $\lambda_1, \lambda_2, ..., \lambda_H$ denote the H labelings of E. The problem of bicluster ensemble is to combine the H biclustering solutions to obtain a biclustering solution $\hat{\pi}$ that achieves one or more of the following aims:

- It improves the quality of the biclusters.
- It is more robust and stable than its constituent schemes.

3. Related work

Hartigan [7] was the first one to coin the term biclustering but it was Cheng and Church [1] who used biclustering for the gene expression data for the first time. Since then, many biclustering algorithms have been developed and used to extract biclusters from gene expression data [1] [2] [3] [4] [5] [23]. A comparative study of different biclustering methods for gene expression data has been done in [4] [24].

Bagging and boosting [25] [26] [27] are standard techniques to obtain ensembles in the area of classification and clustering. Sampling techniques were used to generate individual partitions in the adaptive clustering ensemble technique proposed in [12]. Hypergraph Partitioning [14] [28], Relabeling and voting [25] [15] [27] Co-association based functions [10] [11] and Expectation Maximization [12]are some of the techniques used for the consensus generation. Authors in [29] have proposed ensemble technique using neighborhood search (ENS) for finding quality biclusters in the gene expression data. Yin et al. in their paper [30] have used spectral clustering for ensemble generation. A survey of consensus functions can be found in [31]. Various clustering ensemble techniques have been studied in [32].

Co-clustering and projective clustering are problems that are related to biclustering. Though researchers sometimes claim that all three are same but generally solutions for co-clustering do not allow clusters to overlap on objects and features whereas solutions for projective clustering allows overlapping of features but not of objects. Wang et al. [33] presented an ensemble solution for co-clustering wherein they extract block- constant biclusters generalizing the grid-style partitions to allow different resolutions in different parts of the data matrix. A pair of biclusters may overlap on objects or on features but not on both at the same time. Gullo et al. [34] [35] [36] modeledthe ensemble problem for coclustering/ projective clustering as an optimization problem. However, in projective clustering, an object may belong to more than one biclusters but thetotal sum of the membership is one thereby meaning that if an object completely belongs toone bicluster it does not belong to any other. They project the clusters on one dimension in a fuzzy way. Biclustering is different from these problems/solutions wherein an object/feature may have a total membership more than one and a bicluster is defined by more than one feature. Also, bicluster may overlap both on objects

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

and features simultaneously.

In our earlier work [21], we proposed the use of optimization technique to generate ensemble for biclustering solutions. The consensus is obtained by minimizing thedissimilarity between the obtained biclusters and the aligned input biclusters. Hanczar andNadif in [16] proposed the use of bagging to improve the performance of biclustering schemes. In [18], work in [21] has been extended to show that BiETopti outperforms [16] both in terms of quality as well as time. In another paper, Hanczar and Nadif have used triclustering [37] to form bicluster ensemble. The algorithm suffers from the anomaly that it does not fare well when the input schemes contain true biclusters. Moreover, it looks at local minima whereas the global minima could be far off. They have given a graph showing that the loss function is non-increasing. However, these values of loss function areabsolute rather than relative. Also, the paper discusses that absolute loss function may lead to a condition wherein all feature or examples be removed, and proposes the use of relativevalues instead. On the other hand, values of relative loss function may not necessarily be non-increasing. The algorithm also requires the input schemes to contain equal number ofbiclusters and is compute intensive.

4. BiETclassi: Bicluster Ensemble using classifiers

In this section, we present our first approach for bicluster ensemble. We use the relabeling and voting approach to generate the consensus. Relabeling is done twice, once to align similar biclusters using label correspondence and second time, it is done using classifiers like Discriminant Analysis (DA) and Support vector Machine (SVM) [19]. DA and SVM are basically binary classifiers that work for two classes. However, in gene expression data,genes may be responsible for more than two functions. Classifiers that can handle multipleclasses need to be used instead. Extensions of DA and SVM that solve the multiclass problem are known to exist in literature [38], but they do not allow the classes to overlap. To be ableto handle overlapping biclusters, one needs to consider the multilabel classification [39]. Classifiers that handle multilabel and multiclass also exist in literature [40] [41] but they cannot be directly applied for our problem as they work on the same set of conditionsfor all the labels. On the other hand, in gene expression data, different samples/attributes define different biclusters. Thus, we extend these techniques to suit the need of biclusters.

4.1 The Approach

Our algorithm works in four phases. First two phases generate the schemes and align similar biclusters of different schemes. These two phases are same as that in BiETopti [21] [18] and we briefly recap them below. In phase-III, a classifier is used to predict labels for genes for each input scheme. This is the main contribution of our approach and we discussit in section 4.1.1 in detail. Having predicted the gene labels for each scheme, voting is used to obtain the final consensus in phase-IV. Voting is also used to obtain labels of the conditions. Figure 1 shows the basic architecture of our algorithm.

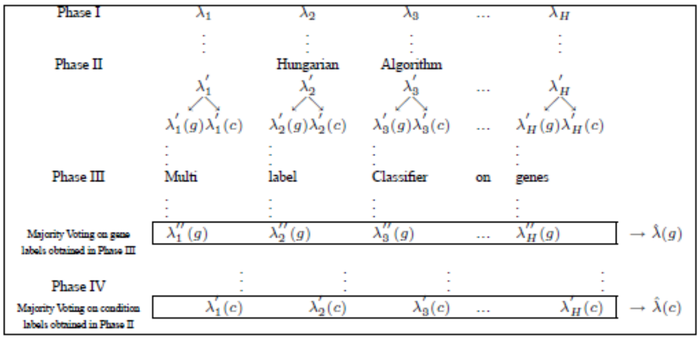


Figure 1: Architecture of BiETclassi

Phase I deals with the generation of input schemes. Various ways in which input schemes can be generated are: by running different algorithms, by running the same algorithm multiple times with random initialization, by varying input parameters or by data re sampling etc. In our approach schemes are generated by running a biclustering algorithm several times with different initializations. In the absence of ground truth, similar biclusters in two different schemes may be assigned different labels. Thus, in phase II, biclusters of two schemes are relabeled and aligned so that similar biclusters in two schemes have the same label. The p-measure defined in [21] [18] is used to find similarity between the biclusters. Hungarian method is then used to relabel the biclusters.

4.1.1 Phase III: Relabeling the genes using a Classifier

Statistical techniques like DA and SVM have been widely used to distinguish classes in multi-attribute data. They classify the objects by identifying the attributes that best define the objects in a group and distinguish them from the remaining objects that are not the member of the same group. The challenge with using classifiers to predict the labels of gene for the biclustering problem is threefold. One, the biclusters are overlapping. Secondly the biclusters are non-exhaustive i.e., there may be genes/samples that do not belong to any bicluster. Thirdly, different biclusters are defined by a different set of attributes/samples; the third being the most important.

Multi-label classifiers are used to address the first problem. In multi-label classification, an object may belong to more than one class. It is different from multiclass classification, wherein objects may be categorized into more than two classes but an object belongs to one class only. We present a method to extend binary classifiers DA and SVM to handle multiclass and multilabel data for biclustering. There are broadly two ways of handling multi label classification [39] [42]. The first being the problem transformation and second being the algorithm adaptation. In problem transformation, the multi-label problem is transformed into a set of binary classification problems, which can then be easily handled. In algorithm adaptation, algorithms are adapted to directly perform multi-label classification instead of transforming the problem. Various problem transformation methods exist in literature. We have used the Binary Relevance method wherein one binary classifier is trained for each label. For multiclass classification we extend one-against-all classification methods for biclustering. For each label (bicluster), a binary class problem is built so that the genes associated with that label are in one class and the rest are in another class. The genes in the bicluster are given the label of the bicluster and the rest of the genes are given the label 0. For each binary class problem, a different set of features corresponding to the conditions of the bicluster is used to take care of the third challenge. Finally, a gene is assigned the union of all the labels. This allows us to assign more than one labels to a gene. This takes care of the overlapping nature of the biclusters. Also there may be genes with special label 0. Such genes do not belong to any bicluster. One against all classification method involves training a single classifier per class, with the objects of that class as positive objects and all other objects as negatives. Figure 2 explains the method pictorially.

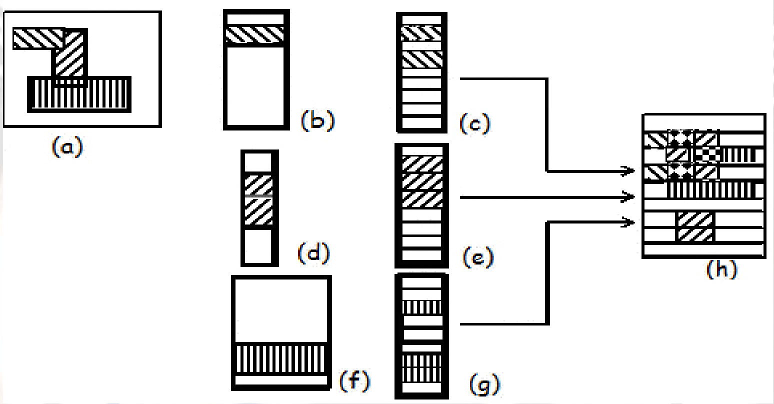
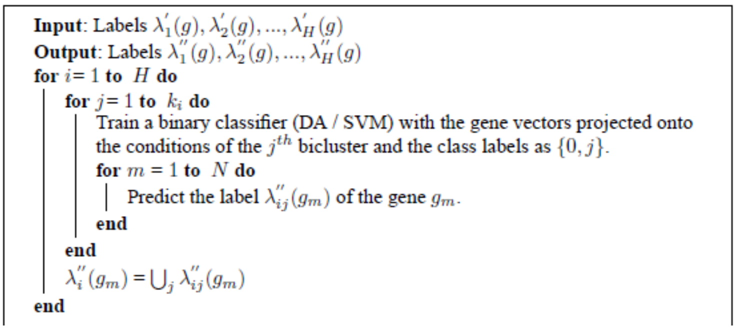


Figure 2: Visualization of one against all method. (a) Biclusters of an input scheme (b)Subject the first bicluster to the classifier on reduced set of conditions/attributes (c) new labels predicted by the classifier for first bicluster (d) Subject the second bicluster to the classifier on reduced set of conditions/attributes (e) new labels predicted by the classifier for second bicluster (f) Subject the third bicluster to the classifier on reduced set of conditions/attributes (g) new labels predicted by the classifier for third bicluster (h) union of the new labels obtained for the genes in (c),(e) and (g)

Consider a bicluster, we apply the standard binary classifier (DA/SVM) on the entire set of genes and the subset of conditions in the bicluster. The genes in the bicluster are given the label of the bicluster and the rest of the genes are given the label 0. The process is repeated for every bicluster of one scheme except the one with special label 0 (genes not belonging to any bicluster in a scheme) and finally the union of all the labels is taken to obtain multiple labels λ'' (g) for a gene g. This is repeated for every scheme. Algorithm 1 summarizes the computation of λ'' (g) for the ith scheme, i = 1 . . . H.



```
Input: Labels $\lambda_1'(g), \lambda_2'(g), ..., \lambda_H'(g)$
Output: Labels $\lambda_1''(g), \lambda_2''(g), ..., \lambda_H''(g)$
for $i = 1$ to $H$ do
    for $j = 1$ to $k_i$ do
        Train a binary classifier (DA / SVM) with the gene vectors projected onto
        the conditions of the $j^{th}$ bicluster and the class labels as $\{0, j\}$.
        for $m = 1$ to $N$ do
            Predict the label $\lambda_{ij}''(g_m)$ of the gene $g_m$.
        end
    end
    $\lambda_i''(g_m) = \bigcup_j \lambda_{ij}''(g_m)$
end
```

Algorithm 1: Predicting gene labels using a Multi-Label Classifier

4.1.2 Phase IV: Final consensus

Finally, voting is used on λ" (g) to generate the final consensus labeling λ^(g) for the genes. A label is assigned to a gene g in the final ensemble if at least τ number of schemes assign the label to g. Similarly, to obtain the final labeling λ^(c) for the conditions, voting is used on the conditions of the aligned biclusters.

5. Experimental Results: BiETclassi

We implemented our algorithm in MATLAB version 7.10 (R2010a) on Intel Core i5- 2430M CPU @2.40 Ghz with 4GB RAM using Windows 7 Home Basic Operating System. Experimental results on synthetic and real datasets are presented in subsequent subsections.

5.1 Results on Synthetic Datasets.

We performed experiments on the benchmark datasets (DS1 and DS2) of Prelic etal [4] using Iterative Signature Algorithm (ISA) [2] as the biclustering algorithm. DS1 and DS2 are two distinct datasets: one having overlapping biclusters without noise and the other nonoverlapping dataset with noise added to the data. The details of synthetic datasets are given in Table 1.

Code-Dataset	Size(N*d)	# implanted biclusters(k)
DS1- Prelic (without noise)	110*110	11(overlapping)
DS2- Prelic (with noise)	100*50	10(non overlapping)

Table 1 Synthetic Datasets

In order to validate the quality of biclusters obtained two statistical measures namely Biclustering Error (BCE) and Average Rand Index (AvRI) have been used. The number of misclassified values ((gene, condition) pairs) in the biclusters amounts to the biclustering error. Misclassified values result from the (gene, condition) pairs that do not match with the class they ought to be in the synthetic dataset. AvRI is same as that defined in [21] [18] to quantify the statistical significance of biclustering solutions.

Input schemes were generated by running ISA on the expression data 20 times, each time with 100 gene seed vectors. The schemes were preprocessed to remove the biclusters with high overlap (> 80%). Ensemble code was then executed. The whole procedure is repeated 20 times and the results are averaged over the runs. As the number of biclusters returned by ensemble is same as that of the reference scheme, we choose the scheme with the largest number of biclusters as the reference scheme, in order not to miss any bicluster. Two sets of experiments were conducted on synthetic datasets of Prelic et al. In the first set, the thresholds (tg, tc) were fixed and the schemes were generated by running ISA with different gene seed vectors. In the second set of experiments, tg was varied keeping both the gene seed vector and tc fixed. The value of tg was varied from [2.4, +2.0] in steps of 0.2. It was observed that for tg values ranging from [0.6, 1.6] schemes with biclusters identical to the implanted biclusters were obtained whereas schemes obtained for tg varying from [2.4, 0.8] biclusters consisted essentially of all the genes and all the biclusters eventually reduced to a single bicluster after preprocessing. Thus, we focused our study on tg varying from [0.6, 0.4] and [1.8, 2.0].

Experiments were performed on dataset DS1 with varying values of threshold for the voting step. The results at varying threshold are shown in Table 2. It was found that the results improved with the increase in the threshold value, however it tends to decrease after a threshold value 80%. So, we fixed the threshold value for voting at 80% for the rest of the experiments.

tg, tc	**50%**	**60%**	**70%**	**80%**	**90%**
-.5,2	.835	.84	.845	.85	.80
-.4,2	.84	.84	.84	.845	.801
-.35,2	.96	.96	.96	.977	.886
1,1	.729	.73	.733	.735	.693
0,1	.56	.565	.565	.565	.503

Table 2 Effect of different voting threshold values on AvRI on DS1

Schemes	**Best**	**BiETopti**	**BiETclassi**	
t_g, t_c ↓			**BiET SVM**	**BiET DA**
-0.50 , 2	3402	2540	2508	2489
-0.40 , 2	3830	3002	3002	2981
-0.35 , 2	3618	2652	2562	2087
1 , 1	5218	3580	3173	3156
0 , 1	5860	3768	3721	3712
vary t_g	3180	2752	2527	2518

(a)

Schemes	**Best**	**BiETopti**	**BiETclassi**	
t_g, t_c ↓			**BiET SVM**	**BiET DA**
-0.50 , 2	.82	.822	.826	.834
-0.40 , 2	.779	.799	.799	.837
-0.35 , 2	.901	.912	.923	.954
1 , 1	.695	.701	.731	.735
0 , 1	.549	.563	.566	.566
vary	.811	.825	.844	.85

(b)

Table 3 BiETclassi on DS1 using (a)BCE and (b)AvRI respectively

Table 3 compares the BCE and AvRI of the best input schemes and that of the biclusters produced by BiETclassi on DS1. The best was computed from the 400 (20 * 20) schemes. The biclusters produced by BiETclassi are also compared with those produced by BiETopti. The values shown are the average of the values obtained in the 20 runs of each experiment. Table shows the results for both sets of experiments. First five rows show results for first set of experiment i.e., when the schemes are generated by varying seed vectors for fixed (tg, tc) values and the last row shows result for the second set of experiments i.e., when the schemes are generated by varying tg for a fixed seed vector and tc. Results of BiETclassi are shown both with SVM and DA. The following inferences can be drawn from the table:

- BiETclassi improves upon the performance of the best input schemes.
- Quality of biclusters produced by BiETclassi is better than BiETopti.
- BiETDA performs better than BiETSVM.

Effect of noise: Noisy dataset (DS2) of Prelic et al. was used to study the impact of noise on the performance of BiETclassi. Table 4 shows the results for the both the sets of experiments using BCE and AvRI respectively. The results are also compared with BiETopti. Again, results of BiETclassi are shown both with SVM and DA. The tables show that BiETclassi was able to produce biclusters better than the best of the input schemes even in presence of noise. BiETclassi performed better than BiETopti. Even in presence of noise BiETDA performs better than BiETSVM.

Schemes	**Best**	**BiETopti**	**BiETclassi**	
t_g, t_c ↓			**BiET SVM**	**BiET DA**
.90, 1	2865	2431	2314	2300
1, .5	3012	2650	2660	2592
-.35, 2	4187	3256	3187	2891
vary t_g	2588	2312	2113	1981

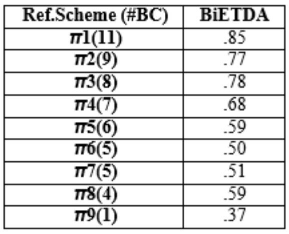
(a)

Schemes	**Best**	**BiETopti**	**BiETclassi**	
t_g, t_c ↓			**BiET SVM**	**BiET DA**
.90, 1	.878	.882	.89	.896
1, .5	.776	.784	.782	.79
-.35, 2	.508	.511	.508	.656
vary t_g	.92	.92	.956	.982

(b)

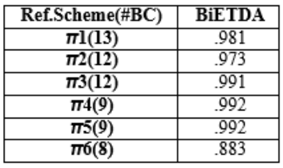
Table 4 BiETclassi on DS2 using (a)BCE and (b)AvRI respectively

Effect of Changing the reference scheme: Table 5 shows the impact of changing the reference scheme on the results. It is evident that the results deteriorate as the number of biclusters in the reference scheme reduces. Last row of the table shows that if a scheme with single bicluster is included and is chosen as a reference, the performance deteriorates drastically. Study on the noisy data shows that the results are best when a reference scheme has number of biclusters close to the number (10) of actual biclusters. Thus, if we have a prior knowledge of the number of biclusters in the dataset, we should choose the scheme with number of biclusters closest to the actual number of biclusters as the reference. Otherwise, we choose the scheme with maximum number of biclusters as the reference scheme.



Ref.Scheme (#BC)	BiETDA
π1(11)	.85
π2(9)	.77
π3(8)	.78
π4(7)	.68
π5(6)	.59
π6(5)	.50
π7(5)	.51
π8(4)	.59
π9(1)	.37

DS1



Ref.Scheme(#BC)	BiETDA
π1(13)	.981
π2(12)	.973
π3(12)	.991
π4(9)	.992
π5(9)	.992
π6(8)	.883

DS2

Table 5 Effect of reference scheme on AvRI on both the datasets DS1 and DS2

5.2 Results on Real Datasets

We also worked on 4 real datasets. The real datasets we used in our study are Saccharomyces cerevisiae (Yeast), Arabidopsis thaliana, Human breast cancer and Diffuse large B-cell lymphoma (DLBCL). Yeast, an organism that is easy to grow is eukaryotic and because of biochemical similar nature as humans, is quite popular with biologists for study purposes. Gene expression behavior during various stress conditions are examined in yeast datasets. Arabidopsis thaliana is a common weed undergoing the same processes of growth, development, flowering etc. as most of the higher plants. It has a small genome and produces a large number of seeds that mature in only about six weeks. To study higher organisms, we also studied two expression datasets of homosapiens, Human breast cancer dataset and Diffuse large B-cell lymphoma dataset. Table 6 gives the details about these datasets.

Organism	Genes(N), Conditions(d)	source
Arabidopsis thaliana	734,69	www.tik.ee.ethz.ch/sop/bicat
Saccharomyces cerevisiae	2993,173	www.tik.ee.ethz.ch/sop/bicat
Diffuse large-B-cell lymphoma	661,180	www.bioinf.jku.at/software/fabia
Human breast cancer	1213,97	www.bioinf.jku.at/software/fabia

Table 6 Real Datasets

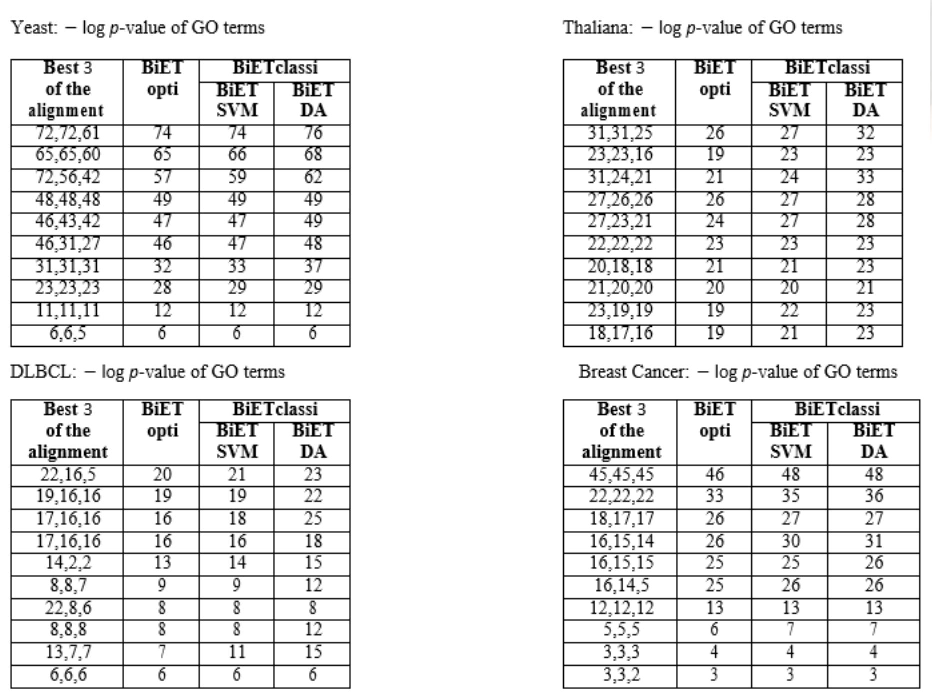
On each of these, input schemes were generated by running ISA, each time with hundred different gene seed vectors. Sizes of the biclusters were kept to be comparable to eliminate the effect of size of biclusters on the p-values.

On real datasets, in the absence of ground truth, we cannot use BCE or AvRI to validate our biclusters. Hence, we use the domain knowledge to determine the biological significance of the biclusters. We validate our biclusters using functional annotation GO (Gene Ontology) and common patterns (motifs) in the promoter regions of the genes of a bicluster with the help of biological tools DAVID and RSAT available on line.

Tables 7 - 8 show the top 10 biclusters obtained from BiETclassi along with their aligned input biclusters which clearly show that there is a huge improvement in the quality of the biclusters obtained. Table 7 shows the GO terms whereas in Table 8 evaluation is done based on motifs. As the tables show, BiETclassi outperforms the best of the input schemes on the real datasets most of the times. Also, the comparison with BiETopti is shown to endorse that BiETclassi outperforms BiETopti.

6. BiETmetaclus: Bicluster Ensemble using Similarity measures

In this section we present our second approach. We have seen in the previous section that algorithm BiETclassi requires input schemes to be aligned. Hungarian algorithm needs to be executed for handling label correspondence. This algorithm also requires a classification problem to be solved. Both of which are expensive algorithm in terms of time. Here we present a technique that does away with the requirement of aligning the schemes and moreover there is no need of solving classification problem. In this approach we form a pool of all the biclusters and form metaclusters consisting of the biclusters with high similarity. Similar technique was used in [16]. However, they use hierarchical clustering to obtain the metaclusters and Jaccard Index to compute the similarity between the biclusters. We hypothesize that a bicluster having high content of information about a group of biclusters is a good representative of all of them. Such a bicluster, share less information with the remaining biclusters.



Yeast: $-\log p$-value of GO terms

Best 3 of the alignment	BiET opti	BiETclassi	
		BiET SVM	BiET DA
72,72,61	74	74	76
65,65,60	65	66	68
72,56,42	57	59	62
48,48,48	49	49	49
46,43,42	47	47	49
46,31,27	46	47	48
31,31,31	32	33	37
23,23,23	28	29	29
11,11,11	12	12	12
6,6,5	6	6	6

Thaliana: $-\log p$-value of GO terms

Best 3 of the alignment	BiET opti	BiETclassi	
		BiET SVM	BiET DA
31,31,25	26	27	32
23,23,16	19	23	23
31,24,21	21	24	33
27,26,26	26	27	28
27,23,21	24	27	28
22,22,22	23	23	23
20,18,18	21	21	23
21,20,20	20	20	21
23,19,19	19	22	23
18,17,16	19	21	23

DLBCL: $-\log p$-value of GO terms

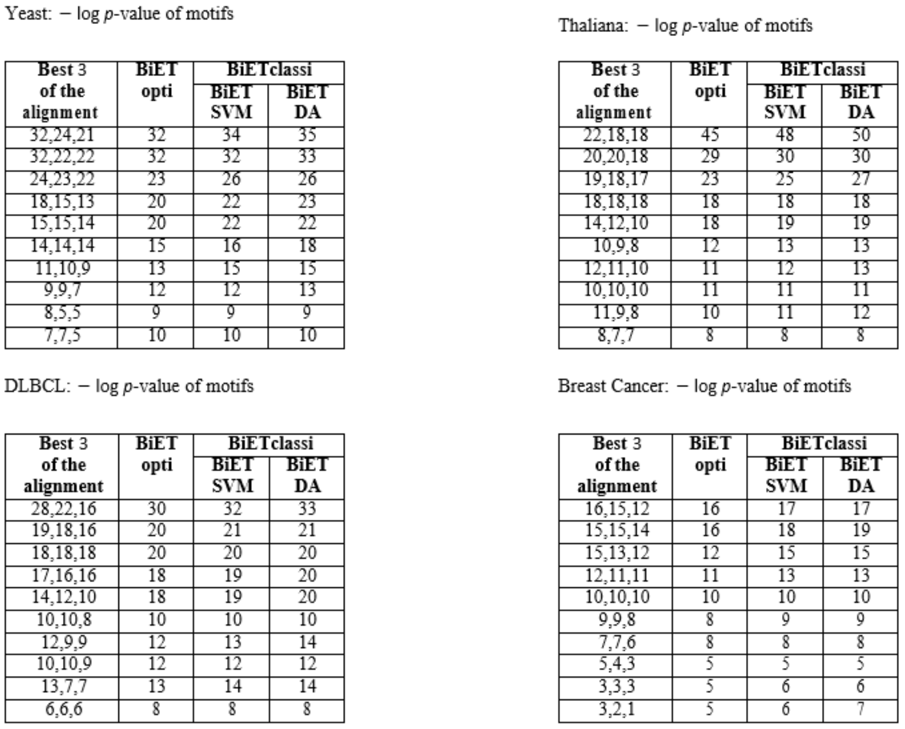
Best 3 of the alignment	BiET opti	BiETclassi	
		BiET SVM	BiET DA
22,16,5	20	21	23
19,16,16	19	19	22
17,16,16	16	18	25
17,16,16	16	16	18
14,2,2	13	14	15
8,8,7	9	9	12
22,8,6	8	8	8
8,8,8	8	8	12
13,7,7	7	11	15
6,6,6	6	6	6

Breast Cancer: $-\log p$-value of GO terms

Best 3 of the alignment	BiET opti	BiETclassi	
		BiET SVM	BiET DA
45,45,45	46	48	48
22,22,22	33	35	36
18,17,17	26	27	27
16,15,14	26	30	31
16,15,15	25	25	26
16,14,5	25	26	26
12,12,12	13	13	13
5,5,5	6	7	7
3,3,3	4	4	4
3,3,2	3	3	3

Table 7: Top 10 biclusters of BiETclassi with best 3 aligned biclusters of the input schemes

Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*



Yeast: − log *p*-value of motifs

Best 3 of the alignment	BiET opti	BiETclassi	
		BiET SVM	BiET DA
32,24,21	32	34	35
32,22,22	32	32	33
24,23,22	23	26	26
18,15,13	20	22	23
15,15,14	20	22	22
14,14,14	15	16	18
11,10,9	13	15	15
9,9,7	12	12	13
8,5,5	9	9	9
7,7,5	10	10	10

Thaliana: − log *p*-value of motifs

Best 3 of the alignment	BiET opti	BiETclassi	
		BiET SVM	BiET DA
22,18,18	45	48	50
20,20,18	29	30	30
19,18,17	23	25	27
18,18,18	18	18	18
14,12,10	18	19	19
10,9,8	12	13	13
12,11,10	11	12	13
10,10,10	11	11	11
11,9,8	10	11	12
8,7,7	8	8	8

DLBCL: − log *p*-value of motifs

Best 3 of the alignment	BiET opti	BiETclassi	
		BiET SVM	BiET DA
28,22,16	30	32	33
19,18,16	20	21	21
18,18,18	20	20	20
17,16,16	18	19	20
14,12,10	18	19	20
10,10,8	10	10	10
12,9,9	12	13	14
10,10,9	12	12	12
13,7,7	13	14	14
6,6,6	8	8	8

Breast Cancer: − log *p*-value of motifs

Best 3 of the alignment	BiET opti	BiETclassi	
		BiET SVM	BiET DA
16,15,12	16	17	17
15,15,14	16	18	19
15,13,12	12	15	15
12,11,11	11	13	13
10,10,10	10	10	10
9,9,8	8	9	9
7,7,6	8	8	8
5,4,3	5	5	5
3,3,3	5	6	6
3,2,1	5	6	7

Table 8: Top 10 biclusters of BiETclassi with best 3 aligned biclusters of the input schemes

Thus, we propose the use of mutual information to compute the metaclusters. We use the concept of well separated seeds to form metaclusters so as to minimize the between group information. To endorse the use of MI as similarity measure, we also used p-measure to form the metaclusters. Metaclusters are obtained by collecting the biclusters with high pairwise mutual information/p-measure in one metacluster. The algorithms are respectively called BiETMI and BiETpM. Voting is then done on the metaclusters to find the final consensus. It is shown in the experimental section that the biclusters produced using MI are biologically better than the other similarity measure, p-measure.

Similarity measure MI used for metacluster generation is discussed in section 6.1. Our approach is given in detail in section 6.2 and the experimental results are discussed in section 7.

6.1 Mutual Information

Mutual Information (MI) between two random variables X and Y is a measure of information contained in X about Y and vice versa. X and Y represent two biclusters if MI is to be calculated between biclusters. If given a value of X, it is easy to predict the value of Y then X contains good amount of information about Y. Clearly, if X and Y are dependent, X and Y can predict each other well and we say that the MI between them is high. And, if X and Y are independent, they cannot predict each other’s behavior and we say that the MI between them is zero. MI is defined as a measure of divergence of the observed joint distribution of X and Y from the hypothesis that X and Y are independent and is given as:

$$I(x,y) = -\sum_{x}\sum_{y} p(x,y)\log\frac{p(x)*p(y)}{p(x,y)}$$

where p (x, y) is the joint probability density function of X and Y, and (x) and p(y) are the marginal probability density functions of X and Y respectively. Mutual information is zero if and only if X and Y are statistically independent i.e., they do not share any information about each other.

Generally, one needs to estimate the distribution as no prior knowledge is available. Two broad classes namely Parametric and Nonparametric are used to estimate the probability distribution functions. Parametric method involves assuming a model for the probability density function and then determining the various parameters from the data. The results are poor if the assumption is poor. In nonparametric approach no assumption about the underlying probability density function is made. Histogram method and Kernel density estimation are two methods of estimating probability density function by the nonparametric approach. We have used MATLAB to compute MI which uses histogram method for estimating the value.

6.2 Our Approach

Our algorithm works in 3 phases. Phase-I for generating the schemes is same as that in the previous section. Meta clusters are formed in Phase-II and this is our main contribution. Voting is done in Phase-III to form a bicluster that represents the consensus of the metacluster.

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

6.2.1 Phase-II Meta Cluster formation

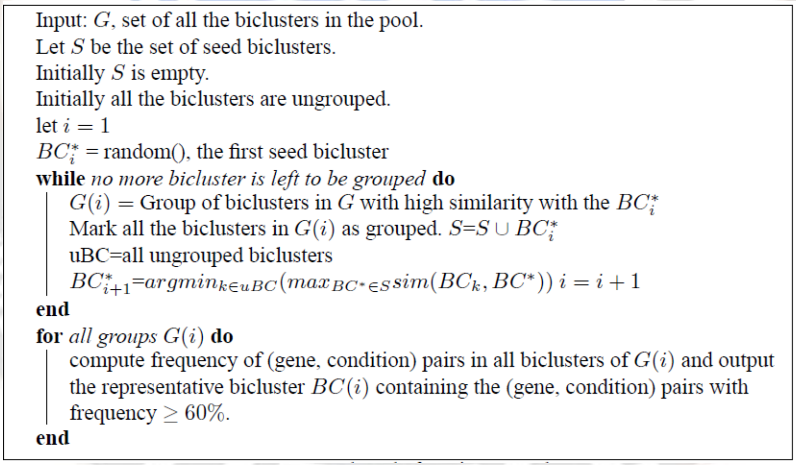
In this step, we collect biclusters of all the schemes in a pool and form groups based on mutual information. Groups, called metaclusters are formed so that they share maximum information (/similarity) within the group and minimum information (/similarity) with the biclusters in other meta-clusters. Thus, metaclusters are formed with the aim to maximize within meta-cluster information (/similarity) and minimize between meta-cluster information (/similarity). To be able to form well separated groups, we construct a set S of seed biclusters. The first seed bicluster BC1 is chosen at random and all biclusters with high mutual information/p-measure with BC1 are grouped together to form one metacluster. Second seed bicluster BC2 is chosen farthest from S i.e., the one that has least mutual information with BC1. Biclusters with high mutual information with BC2 are put in the second metacluster. Next seed bicluster is chosen farthest from S i.e., the one that has least mutual information with both BC1 and BC2. The process of forming metaclusters and selecting a farthest seed bicluster is repeated until no more biclusters are left to be grouped. This method of choosing the seed has also been used in [43] and [44].

In the next phase we will select one representative bicluster for each metacluster. The number of output biclusters is determined by the algorithm itself without requiring the user to specify it.

6.2.2 Phase-III Forming the consensus

Previous phase resulted in the formation of many metaclusters, each having several similar biclusters in it. We need to find a representative of each such metacluster. The bicluster that shares the maximum information with the rest of biclusters in the metacluster is a good candidate for the representative of the group. However, such a representative has a limitation that it has to be one of the biclusters. On the other hand, there may be some (gene, condition) pairs in other biclusters (of the same metacluster) that are important and should have been a part of the final bicluster. Thus, instead, we form the representative bicluster based on frequency of (gene, condition) pairs. Frequency of all (gene, condition) pairs is calculated and these pairs whose value is greater than the threshold η are reported as the elements of the final bicluster. It was experimentally verified that that choosing a representative in this manner is a better alternative than the first method of choosing the representative. Thus, rest of the experiments were performed using this method.

Details of the algorithm are shown in Algorithm 2.



```
Input: $G$, set of all the biclusters in the pool.
Let $S$ be the set of seed biclusters.
Initially $S$ is empty.
Initially all the biclusters are ungrouped.
let $i = 1$
$BC_i^*$ = random(), the first seed bicluster
while no more bicluster is left to be grouped do
    $G(i)$ = Group of biclusters in $G$ with high similarity with the $BC_i^*$
    Mark all the biclusters in $G(i)$ as grouped. $S$=$S \cup BC_i^*$
    uBC=all ungrouped biclusters
    $BC_{i+1}^*$=$argmin_{k \in uBC}(max_{BC^* \in S} sim(BC_k, BC^*))$ $i = i + 1$
end
for all groups $G(i)$ do
    compute frequency of (gene, condition) pairs in all biclusters of $G(i)$ and output
    the representative bicluster $BC(i)$ containing the (gene, condition) pairs with
    frequency $\geq 60\%$.
end
```

Algorithm 2: Pseudocode for BiETmetaclus

7. Experimental Results: BiETmetaclus

Experiments were performed both on synthetic datasets and real gene expression datasets to show the efficiency of our approach. The biclusters produced were compared both with BiETopti and BiETclassi.

7.1 Results on Synthetic Datasets

We performed experiments on the benchmark datasets (DS1 and DS2) of Prelic etal [4]using ISA as the biclustering algorithm as in BiETclassi.

Table 9 compares the performance of BiETmetaclus with the best input schemes and that of BiETclassi and BiETopti on DS1 both in terms of BCE and time. We concluded in the last section that BiETDA performed better than BiETSVM so for the comparison DA version of BiETclassi (BiETDA) is taken. BiETmetaclus is executed using MI (called BiETMI) as the similarity measure for metacluster formation. We also compare the performance of BiETMI with another version of BiETmetaclus that uses p-measure as the similarity measure instead of MI in the metacluster formation step. The values shown in the table are the average of the values obtained in the 20 runs of each experiment. Column 2 gives the best values, of the input schemes, over all the runs. The table shows the results for both the sets of experiments. First five rows of table show the results for first set of experiment i.e., when the schemes are generated by varying seed vectors for fixed (tg, tc) values and the last row shows the result for the second set of experiments i.e., when the schemes are generated by varying tg for a fixed seed vector and tc. Similarly, Table 10 compares the performance of various algorithms in terms of AvRI. The following inferences can be drawn from the tables:

- BiETmetaclus improves upon the performance of the best input scheme both with MI as well as p-measure.
- BiETmetaclus performs better than BiETopti both in terms of quality and time.
- BiETMI performs better than BiETpM.
- BiETmetaclus is faster than BiETclassi.

Observe that BiETclassi performs better than BiETmetaclus. Better performance of BiETclassi comes at the cost of additional time taken by it. Thus, there is a tradeoff between the two approaches as far as quality and time are concerned. Note that with respect to BiETopti, BiETmetaclus improves as regard to quality as well as time.

Schemes t_g, t_c ↓	**Best**	**BiETopti**	**BiETclassi**	**BiETmetaclus**		**Time(sec) BiETopti**	**Time(sec) BiETDA**	**Time(sec) BiETMI**
			BiETDA	**BiETpM**	**BiETMI**			
-0.50 , 2	3402	2540	2489	2540	2540	30.3	18.3	9.9
-0.40 , 2	3830	3002	2981	2998	2990	28.5	17.7	8.9
-0.35 , 2	3618	2652	2087	2562	2426	27	15.6	8.4
1 , 1	5218	3580	3156	3521	3428	51	30.3	23.1
0 , 1	5860	3768	3712	3740	3740	46.7	24.9	13.96
vary t_g	3180	2752	2518	2732	2725	40	22.6	7.6

Table 9 BiETmetaclus on DS1 using BCE

Schemes t_g, t_c ↓	**Best**	**BiETopti**	**BiETclassi**	**BiETmetaclus**	
			BiETDA	**BiETpM**	**BiETMI**
-0.50 , 2	.82	.822	.834	.822	.822
-0.40 , 2	.779	.799	.837	.813	.814
-0.35 , 2	.901	.912	.954	.923	.935
1 , 1	.695	.701	.735	.712	.724
0 , 1	.549	.563	.566	.565	.565
vary t_g	.811	.825	.85	.83	.831

Table 10 BiETmetaclus on DS1 using AvRI

Effect of noise: The performance of BiETmetaclus was also studied on dataset (DS2) of Prelic et al. to see the impact of noise. Table 11 shows the results for both the sets of experiments using BCE and AvRI. Again, results of BiETmetaclus are shown both with MI and p-measure. The tables show that BiETmetaclus was able to extract biclusters better than the best of the input schemes even in presence of noise. Even in presence of noise BiETMI performs better than BiETpM.

Schemes *tg, tc* ↓	**Best**	**BiET opti**	**BiET classi**	**BiETmetaclus**		**Schemes** *tg, tc* ↓	**Best**	**BiET opti**	**BiET classi**	**BiET metaclus**	
			BiET DA	**BiET pM**	**BiET MI**				**BiET DA**	**BiET pM**	**BiET MI**
.90, 1	2865	2431	2300	2412	2412	**.90, 1**	.878	.882	.896	.886	.886
1, .5	3012	2650	2592	2631	2618	**1, .5**	.776	.784	.79	.785	.786
-.35, 2	4187	3256	2891	3203	3195	**-.35, 2**	.508	.511	.656	.523	.54
vary *tg*	2588	2312	1981	2100	2091	**vary** *tg*	.92	.92	.982	.956	.962

(a) (b)

Table 11 BiETmetaclus on DS2 using (a)BCE and (b)AvRI respectively

7.2 Results on Real Datasets

Experimental studies were performed on the real expression datasets with BiETmetaclus. On each of these, we generated input schemes by running ISA each time with hundred different gene seed vectors. Sizes of the biclusters were kept to be comparable to eliminate the effect of size of biclusters on the p-values. Tables 12 - 13 shows the top 10 biclusters of obtained from BiETmetaclus along with their aligned input biclusters which clearly show that there is improvement in the quality of the biclusters obtained. Table 12 shows the comparison based on GO terms whereas table 13 shows the comparison using motifs. The tables show that BiETmetaclus outperforms not only the best of the input schemes but also

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

the biclusters produced by BiETopti. It is observed that BiETclassi is able to produce biologically better biclusters than BiETmetaclus but at the cost of time as it can be clearly seen from Table 14. The time has been shown for the approach BiETclassi with DA as the classifier. For BiETmetaclus, time has been shown wherein MI is used as the similarity measure. The time taken by BiETMI is less than time taken by BiETDA on all the organisms. The tradeoff between time and quality among BiETclassi and BiETmetaclus is observed on real datasets also on all the organisms. BiETmetaclus wins over BiETclassi as far as time is concerned.

Yeast: −log *p*-value of GO terms | Thaliana: −log *p*-value of GO terms

Best 3 **of the input**	**BiET opti**	**BiET classi**	**BiET metaclus**		**Best** 3 **of MI the input**	**BiET opti**	**BiET classi**	**BiET metaclus**	
		BiET DA	**BiET pM**	**BiET**			**BiET DA**	**BiET pM**	**BiETMI**
72,72,61	74	76	74	74	31,31,25	26	32	27	31
65,65,60	65	68	66	67	23,23,16	19	23	23	23
72,56,42	57	62	59	59	31,24,21	21	33	24	25
48,48,48	49	49	49	49	27,26,26	26	28	27	28
46,43,42	47	49	47	48	27,23,21	24	28	27	27
46,31,27	46	48	48	48	22,22,22	23	23	23	23
31,31,31	32	37	35	36	20,18,18	21	23	21	22
23,23,23	28	29	28	29	21,20,20	20	21	20	21
11,11,11	12	12	12	12	23,19,19	19	23	21	21
6,6,5	6	6	6	6	18,17,16	19	23	21	22

DLBCL: −log *p*-value of GO terms | Breast Cancer: −log *p*-value of GO terms

Best 3 **of the input**	**BiET opti**	**BiET classi**	**BiET metaclus**		**Best** 3 **of MI the input**	**BiET opti**	**BiET classi**	**BiET metaclus**	
		BiET DA	**BiET pM**	**BiET**			**BiETDA**	**BiET pM**	**BiETMI**
22,16,5	20	23	22	22	45,45,45	46	48	47	47
19,16,16	19	22	19	19	22,22,22	33	36	35	35
17,16,16	16	25	18	23	18,17,17	26	27	26	26
17,16,16	16	18	16	16	16,15,14	26	31	28	30
14,2,2	13	15	14	15	16,15,15	25	26	25	25
8,8,7	9	12	10	11	16,14,5	25	26	25	25
22,8,6	8	8	8	7	12,12,12	13	13	13	13
8,8,8	8	12	9	9	5,5,5	6	7	7	7
13,7,7	7	15	11	12	3,3,3	4	4	4	4
6,6,6	6	6	6	6	3,3,2	3	3	3	3

Table 12 Top 10 biclusters of BiETmetaclus with 3 best aligned biclusters of the input schemes

Yeast: −log *p*-value of motifs | Thaliana: −log *p*-value of motifs

Best 3 **of the input**	**BiET opti**	**BiET classi**	**BiET metaclus**		**Best** 3 **of MI the input**	**BiET opti**	**BiET classi**	**BiET metaclus**	
		BiETDA	**BiET pM**	**BiET**			**BiETDA**	**BiET pM**	**BiETMI**
32,24,21	32	35	34	34	22,18,18	45	50	48	49
32,22,22	32	33	33	33	20,20,18	29	30	29	29
24,23,22	23	26	24	25	19,18,17	23	27	25	26
18,15,13	20	23	22	22	18,18,18	18	18	18	18
15,15,14	20	22	22	22	14,12,10	18	19	18	19
14,14,14	15	18	17	17	10,9,8	12	13	13	13
11,10,9	13	15	14	15	12,11,10	11	13	12	12
9,9,7	12	13	12	12	10,10,10	11	11	11	11
8,5,5	9	9	9	9	11,9,8	10	12	11	11
7,7,5	10	10	10	10	8,7,7	8	8	8	8

DLBCL: −log *p*-value of motifs | Breast Cancer: −log *p*-value of motifs

Best 3 **of the input**	**BiET opti**	**BiET classi**	**BiET metaclus**		**Best** 3 **of MI the input**	**BiET opti**	**BiET classi**	**BiET metaclus**	
		BiETDA	**BiET pM**	**BiET**			**BiETDA**	**BiET pM**	**BiETMI**
28,22,16	30	33	32	32	16,15,12	16	17	16	16
19,18,16	20	21	20	20	15,15,14	16	19	18	19
18,18,18	20	20	20	20	15,13,12	12	15	15	15
17,16,16	18	20	19	20	12,11,11	11	13	12	13
14,12,10	18	20	19	20	10,10,10	10	10	10	10
10,10,8	10	10	10	10	9,9,8	8	9	9	9
12,9,9	12	14	13	13	7,7,6	8	8	8	8
10,10,9	12	12	12	12	5,4,3	5	5	5	5
13,7,7	13	14	13	13	3,3,3	5	6	5	6
6,6,6	8	8	8	8	3,2,1	5	7	6	6

Table 13 Top 10 biclusters of BiETmetaclus with 3 best aligned biclusters of the input schemes

Organism	**BiETopti Time(sec)**	**BiETDA Time(sec)**	**BiETMI Time(sec)**
Yeast	8200	801	337
A. Thaliana	565	225	165
DLBCL	647	125	86
Breast Cancer	180	155	39

Table 14 Comparison of time to execute BiETopti, BiETDA and BiETMI

8. Comparison of BiETclassi and BiETmetaclus with existing biclustering algorithms and BiETopti

Figure 3 shows the comparison of the biclusters produced by the two ensemble approaches, BiETclassi and BiETmetaclus, with the biclusters produced by existing biclustering algorithms like order-preserving sub matrix (OPSM) [45], Cheng and Church (CC) [1], BIMAX [4] and ISA [2]. For Yeast and Thaliana, the biclusters for all the biclustering algorithms were taken from the BICAT site. For DLBCL and Breast Cancer, biclusters were generated by executing these algorithms in [8] tool. Figure shows that none of the existing algorithms is said to be a clear winner in all the organisms. CC performs best amongst the existing algorithms on Yeast. On A. Thaliana and DLBCL, performance of ISA is best amongst the existing algorithms. OPSM takes the lead in Breast Cancer dataset. BiETclassi and BiETmetaclus outperform the best in each of these organisms except A. Thaliana. Both outperform BiETopti in all the organisms.

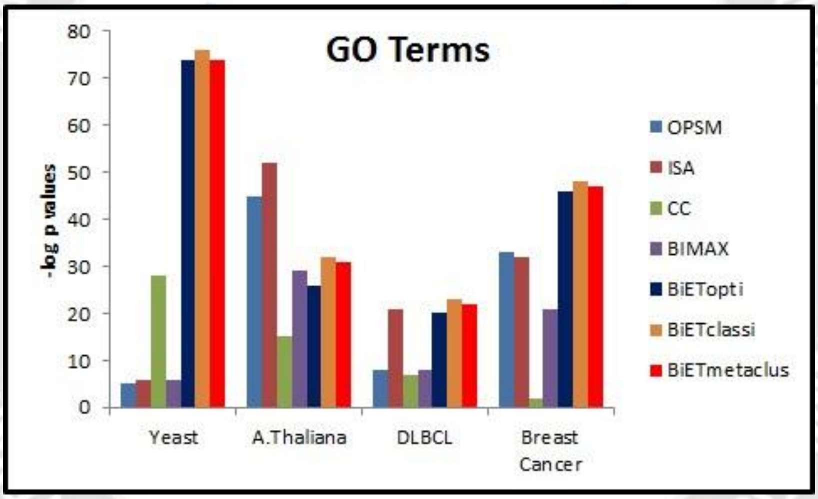


Figure 3: BiETclassi and BiETmetaclus compared with ISA, OPSM, CC, BIMAX andBiETopti

9. Time Analysis

BiETclassi uses classifiers like DA and SVM having the complexity of O (N 3). The classifier is invoked H.k times in the algorithm. Thus, the total time complexity is O (H.k.N 3). BiETmetaclus on the other hand uses similarity measures like Mutual Information and p- measure. The time complexity of BiETmetaclus using MI and p-measure is O ((H. k)2 Nd). BiETopti algorithm takes O ((N + d) k)3.5 time. Ignoring the small constants H and k time complexity of BiETclassi, BiETmetaclus and BiETopti is O (N 3), O(Nd) and O ((N + d)3.5) respectively. The value of d is generally much smaller as compared to N. Thus, BiETmetaclus is much faster than both BiETopti as well as BiETclassi.

10. Conclusion and Future Work

In this paper, we presented two ensemble techniques for the biclustering problem that allows simultaneous overlap of objects and features. Linear classifiers were used to predict labels of genes. Although SVM and DA used in BiETclassi are compute intensive, they give better results than the existing algorithms (HN and BiETopti). Second approach, BiETmetaclus uses similarity measures like MI and p-measure. It also outperforms HN and BiETopti. BiETmetaclus saves in terms of time as compared to BiETclassi whereas quality of biclusters produced by BiETclassi is better than that produced by BiETmetaclus. Thus, there is a tradeoff between the quality of biclusters and the time between BiETclassi and BiETmetaclus. In future we would like to explore more efficient method both in terms of quality and time for the same.

Statements and Declarations

- The authors have no relevant financial or nonfinancial interests to disclose.
- The authors have no competing interest to declare that are relevant to the content of this article.

- All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.
- The authors have no financial or proprietary interests in any material discussed in this article.

REFERENCES

1. Y. Cheng and G. M. Church, "Biclustering of Expression Data," in *Eighth International Conference on Intelligent Systems for Molecular Biology*, La Jolla/ San Diego, CA, USA, 2000.
2. S. Bergmann, J. Ihmels and N. Barkai, "Iterative signature algorithm for the analysis of large-scale gene expression data," *Physical review. E, Statistical, nonlinear, and soft matter physics,* vol. 67, 2003.
3. N. Gupta and S. Aggarwal, "MIB: Using Mutual Information for Biclustering gene expression data," *Elsevier Journal of Pattern Recognition,* pp. 2692-2697, 2010.
4. A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig and E. Zitzler, " A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics,* pp. 1122-1129, 2006.
5. A. Tanay, R. Sharan and R. Shamir, "Discovering statistically significant biclusters in gene expression data," in *ISMB*, 2002.
6. A. Ali, A. Ajil, A. Meenakshi Sundaram and N. Joseph, " Detection of Gene Ontology Clusters Using Biclustering Algorithms," *SN Computer Science,* vol. 4, no. 3, 2023.
7. J. A. Hartigan, "Direct Clustering of a Data Matrix.," *Journal of the American Statistical Association,* vol. 67, pp. 123-129, 1972.
8. S. Barkow-Oesterreicher, S. Bleuler, A. Prelic, P. Zimmermann and E. Zitzler, "BiCAT: A Biclustering Analysis ToolBox," *Bioinformatics,* vol. 22, pp. 1282-1283, 2006.
9. N. K. Verma, T. Sharma, S. Dixit, P. Agrawal, S. Sengupta and V. Singh, "BiDeal: A Toolbox for Bicluster Analysis-Generation, Visualization and Validation," *SN Computer Science*, vol. 2, no. 1, 2021.
10. A. L. N. Fred, "Data Clustering Using Evidence Accumulation," in *Int'l Conference on Pattern Recognition*, 2002.
11. A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation.," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 835-850, 2005.
12. A. P. Topchy, B. M. Bidgoli, A. K. Jain and W. F. Punch, "Adaptive clustering ensembles.," *ICPR,* pp. 272-275, 2004.
13. A. Topchy, A. K. Jain and W. Punch, "A mixture model of clustering ensembles," in *In Proc. SIAM Intl. Conf. on Data Mining*, 2004.
14. A. Strehl and J. Ghosh, "Cluster ensembles – a knowledge reuse framework for combining multiple partitions," *Journal on Machine Learning Research (JMLR), MIT Press,* vol. 3, p. 583–617, 2002.
15. C. Krumpelman and J. Ghosh, " Matching and visualization of multiple overlapping clusterings of microarray data," in *CIBCB*, 2007.
16. B. Hanczar and M. Nadif, "Using the bagging approach for biclustering of gene expression data," *Neurocomputing,* vol. 74, p. 1595–1605, 2011.
17. G. Aggarwal and N. Gupta, "BiETopti-BiClustering Ensemble Using Optimization Techniques," in *ICDM*, 2013.
18. G. Aggarwal and N. Gupta, "BiETopti-BiClustering Ensemble Technique Using Optimization," in *IJBRA*, 2017.
19. R. O. Duda, P. E. Hart and a. D. G. Stork., Pattern Classification, New York: Wiley, 2001.
20. N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods, New York, NY, USA: Cambridge University Press, 2000.
21. G. Aggarwal and N. Gupta, "BEMI: Bicluster Ensemble Using Mutual Information," in *ICMLA*, 2013.
22. I. Priness, O. Maimon and I. E. Ben-Gal, "Evaluation of gene-expression clustering via mutual information distance measure," *BMC Bioinformatics,* vol. 8, 2007.
23. G. I. Allen, E. C. Chi and R. G. Baraniuk., "Convex biclustering," *arXiv:1408.0856 [stat.ME]*, 2014.
24. S. Busygin, O. Prokopyev and P. M. Pardalos., " Biclustering in data mining. Computers & Operations Research," *Special Issue: Bio-inspired Methods in Combinatorial Optimization.,* p. 2964 – 2987, 2008.
25. B. Fischer and J. M. Buhmann, " Bagging for path-based clustering," *IEEE Trans. Pattern Anal. Mach. Intell., Washington, DC, USA, IEEE Computer Society.,* vol. 25, pp. 1411-1415, 2003.
26. J. V. Moreau and A. K. Jain, "The bootstrap approach to clustering.," in *NATO Advanced Study Institute on Pattern recognition theory and applications Springer-Verlag.*, London, UK, 1987.
27. Fridlyand, "Bagging to improve the accuracy of a clustering procedure.," *Bioinformatics,* vol. 19, p. 1090–1099, 2003.
28. X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *ICML*, 2004.
29. B. S. Biswal, M. Anjali and V. Swati, "Ensemble Neighborhood Search (ENS) for biclustering of gene expression microarray data and single cell RNA sequencing data," *Journal of King Saud University - Computer and Information Sciences,*, vol. 34, p. 5, 2022.
30. L. Yin and Y. Liu, " Ensemble biclustering gene expression data based on the spectral clustering.," *Neural Computing and Applications*, 2018.
31. R. Ghaemi, N. Sulaiman, H. Ibrahim and N. Mustapha, " A survey: Clustering ensembles techniques.," in *World Academy of Science, Engineering and Technology*, 2009.
32. S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *IJPRAI*, vol. 25, no. 3, pp. 337-372, 2011.

33. P. a. L. K. a. D. C. a. J. M. Wang, "Nonparametric Bayesian Co-clustering Ensembles," in *Proceedings of the 11th SIAM International Conference on Data Mining*, 2011.
34. F. Gullo, A. K. M. K. A. Talukder, S. Luke, C. Domeniconi and A. Tagarelli, "Multiobjective optimization of co-clustering ensembles," in *Genetic and Evolutionary Computation Conference*, Philadelphia, USA, 2012.
35. F. Gullo, C. Domeniconi and A. Tagarelli, "Enhancing Single-Objective Projective Clustering Ensembles," in *International Conference on Data Mining, IEEEE Computer Society*, Sydney, Australia, 2010.
36. F. Gullo, C. Domeniconi and A. Tagarelli, "Projective Clustering Ensembles," in *The Ninth IEEE International Conference on Data Mining*, Miami Florida USA, 2009.
37. B. Hanczar and M. Nadif, "Ensemble methods for biclustering tasks. Pattern Recognition," vol. 45, p. 3938 – 3949, 2012.
38. C. M. Bishop., Pattern Recognition and Machine Learning, Springer , 2006.
39. G. Tsoumakas and I. Katakis, " Multi-label classification: An overview," *Int J Data Warehousing and Mining,*, p. 1–13, 2007.
40. M. l. Zhang and Z. h. Z. Ml-knn, "A lazy learning approach to multi-label learning.," *PATTERN RECOGNITION,* vol. 40, 2007.
41. M. l. Zhang and Z. h. Zhou, "Multi-label neural networks with applications to functional genomics and text categorization," 2005.
42. M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on .Knowledge and Data Engineering,* no. , p. 99, 2013.
43. N. Gupta and S. Aggarwal, " SISA: Seeded Iterative Signature Algorithm for Biclustering Gene Expression Data," in *IADIS European Conf. Data Mining,* 2008.
44. C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu and J. S. Park, "Fast algorithms for projected clustering," in *SIGMOD Conference*, 1999.
45. A. Ben-Dor, B. Chor, R. M. Karp and Z. Yakhini, "Amir Ben-Dor, Benny Chor, Richard M.Discovering local structure in gene expression data: The order-preserving submatrix problem," *Journal of Computational Biology,* pp. 373-384, 2003