International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 1 Issue 4, June– 2017, Pages: 28-37

Synchronized Clustering: A review on Systematic comparisons and Validation of prominent Block-clustering Algorithms

Dr.Mani Sarma Vittapu
Assistant Professor, Centre of ITSC,
Addis Ababa Institute of Technology,
Addis Ababa University,
Addis Ababa, Ethiopia

***Abstract**: One of the major problems in clustering is the need of specifying the optimal number of clusters in some clustering algorithms. Numerous indices were proposed in order to find reasonable number of clusters. The purpose of the paper is to test the performance and ability of some indices to detect the proper number of clusters on rows and columns partitions obtained by a block clustering algorithm. Simultaneous clustering methods perform clustering in the two dimensions simultaneously. The purpose of the paper is to test the performance and ability of some indices to detect the proper number of clusters on rows and columns partitions obtained by a block clustering algorithms .and also focus on a large number of existing simultaneous clustering approaches applied in text mining, web mining, information retrieval as well as bioinformatics and categorize them in accordance with the methods used to perform the clustering and the intention applications. The main goal of this paper is to provide a systematic comparison and validation of prominent bi-clustering algorithms.*

Keywords: Simultaneous clustering, Bi-clusters, Block clustering, local and global model, constant value bi-cluster, coherent value bi-cluster, coherent evaluation bi-cluster and indices

1. INTRODUCTION

Simultaneous clustering, usually designated by biclustering, co-clustering or block clustering, is an important technique in two way data analysis. The goal of simultaneous clustering is to find submatrices, which are subgroups of rows and subgroups of columns that exhibit a high correlation. A number of algorithms that perform simultaneous clustering on rows and columns of a matrix have been proposed to date. They have practical importance in a wide variety of applications such as biology, data analysis, text mining and web mining. A wide range of different articles were published dealing with different kinds of algorithms and methods of simultaneous clustering. Comparisons of several biclustering algorithms can be found, One of the major problems of simultaneous clustering algorithms, similarly to the simple clustering algorithms, is that the number of clusters must be supplied as a parameter. To overcome this problem, numerous strategies have been proposed for finding the right number of clusters. Several measures for validation exist in clustering area, but they are usually not applied for bi-clustering methods for Validation and comparison are made by external indices. Non-biological indices as sensitivity and specificity are used when information of clustering is known, usually in synthetic data where biclusters are embedded. Only constant and additive biclusters are treated, as they are the most extended. Biological indices are used when no information intrinsic to the data is known. Internal and relative indices are seldom used because biclustering concepts are hard to adapt to clustering indices.

2. SIMULTANEOUS CLUSTERING APPROACH

Given the data matrix A, with set of rows X = (X1, ..., Xn) and set of columns Y = (Y1, ..., Yn), aij , $1 \leq i \leq n$ and $1 \leq j \leq n$ is the value in the data matrix A corresponding to row i and column j. Simultaneous clustering algorithms aim to identify a set of biclusters Bk(Ik, Jk), where Ik is a subset of the rows X and Jk is a subset of the columns Y. Ik rows exhibit similar behaviour across Jk columns, or vice versa and every bicluster Bk satisfies some criteria of homogeneity

Table 1. Comparison between Clustering and Simultaneous clustering

Clustering	**Simultaneous Clustering**
applied to either the rows or the columns of the data matrix **separately** **global model**.	performs clustering in the two dimensions **simultaneously** **local model**.
produce **clusters** of rows **or** clusters of columns.	seeks **blocks** of rows **and** columns that are interrelated
- Each subject in a given subject cluster is defined using **all** the variables. Each variable in a variable cluster characterizes **all** subjects.	Each subject in a bicluster is selected using **only** a subset of the variables and each variable in a bicluster is selected using **only** a subset of the subjects.

Clusters are **exhaustive**	The clusters on rows and columns **should not be exclusive** and/or **exhaustive**

2.1 Bicluster classification

A bicluster can be defined as 'a subset of objects (rows or columns) that jointly respond across a subset of other objects (columns or rows)'. In bioinformatics, rows usually refer to genes and columns to experiments or organism conditions. Madeira and Oliveira [10] classify biclusters depending on what is considered for 'jointly responds':

– Constant value bicluster (*C*): all elements have exactly the same value (μ). Elements of constant bicluster $B = [bi\ j\ ]$ with *n* rows and *m* columns are defined as $bi\ j = \mu$ ---------- (1)

Coherent value bicluster (*H*): row and/or column variations are somehow related. This relationship may be additive (*H*+), multiplicative (*H*×) or by sign (*H* • }). In case of *H*+ and *H*×, each row and/or column differs from others in an additive or multiplicative factor (eqs. 2 and 3, respectively). In case of *H* • }, it is just a qualitative rule of change in tendency (α and β are binary vectors representing increasing or decreasing respect to another row or column –such as 1 or -1–, but it's not imposed any quantitative restriction on $ri\ j$, $ci\ j$ variations)

$$b_{ij} = \mu + \alpha_i + \beta_j \qquad (2)$$

$$b_{ij} = \mu \alpha_i \beta_j \qquad (3)$$

$$b_{ij} = (b_{(i-1,j)} + \alpha_i r_{ij}) + (b_{(i,j-1)} + c_{ij}\beta_j) \qquad (4)$$

– Coherent evolution bicluster (*E*): expression levels are first mapped to labels under certain criteria, such as order or proximity

The above definitions can be applied to rows, columns or both, but measures are usually used in both dimesions. *C* biclusters are almost ideal, so algorithms searching for *C* biclusters usually treats 'constant' as a range of near values by a mapping with coherence evolution. This bicluster classification presents overlaps. For example, *C* biclusters on rows and columns (*Crc*) are included in *C* biclusters on rows (*Cr*)and*C* biclusters on columns (*Cc*). *C* biclusters of any type are included in *H*+ biclusters and overlap with *H*× biclusters. *H* • } includes them all (Fig. 1). This will be important when comparing biclustering algorithms that search for different kinds of biclusters.

C is the most used group because of direct interpretation in biological data. *H*+ biclusters, representing more subtle relations in data are the second group in references. *H*× and *H* • } are rarely used, being their biological relevance difficult to justify or interpret

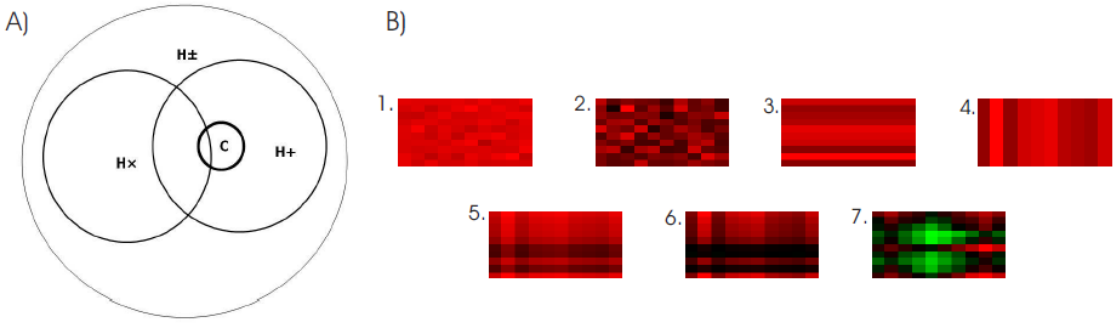

Fig. 1. A) Bicluster sets. Each of the sets is internally divided in row, column and both dimensions biclusters of the corresponding type. B) Heatmaps of different biclusters: 1) C_{rc} bicluster, 2) C_{rc} bicluster with high noise, 3) C_r bicluster, 4) C_c column constant, 5) H^+ bicluster, 6) $H^\times$ bicluster and 7) $H^\pm$ bicluster. 5),6) and 7) become, after row/column transformation, C_r and/or C_c biclusters 3) and 4).

2.2 Coherence measures

Having in mind the different groups of biclusters, we can define measures that determine
how constant or how (additive, multiplicative, sign) coherent is our bicluster. Biclustering algorithms define internally what is considered coherent, but not always under an specific measure or value. Coherencemeasures can be used to define synthetic biclusters for testing or to check if the results over real data fits the bicluster definition of the algorithm. Constancy by rows of bicluster B ($Cr(B)$) and by columns ($Cc(B)$) are easy to measure by means of Euclidean distance

$$C_r(B) = \frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sqrt{\sum_{k=1}^{m}(b_{ik}-b_{jk})^2} \tag{5}$$

$$C_c(B) = \frac{1}{m}\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\sqrt{\sum_{k=1}^{n}(b_{ki}-b_{kj})^2} \tag{6}$$

Overall constancy $C_{rc}(B)$ can be derived from $C_r(B)$ and $C_c(B)$:

$$C_{rc}(B) = \frac{nC_r(B)+mC_c(B)}{n+m} \tag{7}$$

The average measure for all the biclusters found by an algorithm is the weighted mean of the measure for each bicluster. These measures, traditionally used to determine cluster compactness will give bad scores for coherent biclusters. To measure coherency, an incremental treatment of the data can be applied to make them 'constant', then applying above formulas to the transformed bicluster $B' = [b'i\,j]$. In case of $H+$:

$$b'_{ij} = b_{ij} - b_{(i-1)j}\ (b_{0j} = 0) \tag{8}$$

$$b'_{ij} = b_{ij} - b_{i(j-1)}\ (b_{i0} = 0) \tag{9}$$

That way, as seen in Fig. 1b, $H+$ bicluster becomes Cr and/or Cc bicluster, and can be measured by eqs. 5, 6 and 7. A similar transform can be done with $H\times$ using division instead of subtraction, but now there is necessary to include an exception to avoid divisions by zero:

$$b'_{ij} = b_{ij}/b_{(i-1)j}\ (b_{0j} = 1) \quad (10)$$

$$b'_{ij} = b_{ij}/b_{i(j-1)}\ (b_{i0} = 1) \quad (11)$$

Finally, $H^{\pm}$ has a similar treatment:

$$b'_{ij} = 1 \Leftrightarrow b_{ij} > b_{(i-1)j},\ b'_{ij} = -1\ otherwise.\ (b_{0j} = 1) \quad (12)$$

$$b'_{ij} = 1 \Leftrightarrow b_{ij} > b_{i(j-1)},\ b'_{ij} = -1\ otherwise.\ (b_{i0} = 1) \quad (13)$$

Proximity to zero on all these measures points that the bicluster has the corresponding coherence property. There is no limit in the value they can take, but values above 1.5 usually tell us that coherency is lost (see Section 4 for some practical cases).

Table 2. Simultaneous clustering algorithms

Algorithm	Application	Approach	Data type
Two-way splitting [20]	Other	Variance minimization	Continuous
CROEUC [17]	Other	Two-way clustering	Continuous
CROKI2 [17] [9]	Other	Two-way clustering	Categorical
CROBIN [17]	Other	Two-way clustering	Binary
CTWC [15]	Bioinformatique	Two-way clustering	Continuous
Plaid Models [22]	Bioinformatique	Probabilistic and generative	Continuous
δ-biclusters [10]	Bioinformatique	Variance minimization	Continuous
δ-ks patterns [7]	Bioinformatique	Variance minimization	Continuous
ITWC [32]	Bioinformatique	Two-way clustering	Continuous
DCC [5]	Bioinformatique	Two-way clustering	Continuous
OPSM [4]	Bioinformatique	Motif and pattern recognition	Continuous
SAMBA [31]	Bioinformatique	Probabilistic and generative	Continuous
FLOC [36]	Bioinformatique	Variance minimization	Continuous
Spectral [37]	Bioinformatique	Motif and pattern recognition	Continuous
IT [13]	Text Mining	Probabilistic and generative	Continuous
BSGP [12]	Text Mining	Bi-partite Graph	Categorical
cHawk [1]	Bioinformatique	Bi-partite Graph	Continuous
[30]	Other	Probabilistic and generative	Categorical
Block-EM [16]	Other	Two-way clustering	Continuous binary
Block-CEM [16]	Other	Two-way clustering	Continuous binary
Cemcroki2 [26]	Other	Two-way clustering	Categorical

Probabilistic and generative methods use model-based techniques to define biclusters . Probabilistic Relational Models (PRMs) and their extension ProBic are fully generative models that combine probabilistic modeling and relational logic. cMonkey is a generative approach which models biclusters by Markov chain processes. Gu and Liu generalized.
The plaid models proposed in to fully generative models called Bayesian BiClustering model (BBC). The latter models introduced in and are generative models which have the advantage that they select models using well-understood model selection techniques such as maximum likelihood.

3. VALIDATION INDICES

Clustering validation indices are divided into three categories: external, internal and relative. External indices measure the similarity between clustering results and a priori knowledge. Internal indices compare the intrinsic structure of data with cluster results. Internal indices are much harder to apply to biclustering than external indices because much of the internal concepts (such as compactness or separation) are not applying to biclusters, where overlapping and coherent variations are usual. Finally relative indices compare different configurations of input parameters and cluster results, trying to find optimal or stable parameters for a given input data. In the context of biclustering, external validation is mainly used, preferring biological indices to traditional ones. Internal and relative indices are seldom used, because of the non trivial task of adapting biclustering concepts as overlapping and bi-dimensionality to clustering indices.

3.1 Biological external indices

Biological knowledge used in validations are usually gene annotations as those of Gene Ontology (GO) or KEGG. We will call them external indices because imply information external to the data. Given a bicluster *B*, we get all (in example) GO terms annotated to any of the genes in *B* and then apply a statistical significance test to determine if each term appearance is relevant. Biclustering algorithms presented in use GO and/or KEGG enrichment. Other biological knowledge applied in the same way than annotations is related with Transcription Regulatory Networks (TRNs). A TRN is a directed acyclic graph where nodes are genes, and an edge between gene *A* and gene *B* means that gene *A* encodes for a transcription factor protein that transcriptionally regulates (activate or repress) gene *B*. In this case it is considered the number of genes connected in our bicluster or the average distance between genes in it . It's expected that the number of genes connected will be greater and the average distance lower than in random biclusters, which is checked with a significance test. Another interesting characteristic to check is the number of network motifs (substructures that appear in TRNs that are included in a bicluster, but it is seldom used in bibliography.
Although useful for the objective of knowledge discovery, biological significance has a major disadvantage as a validation method: biological knowledge is not complete. When a bicluster does not group known GO/KEGG annotations, or connected genes in a TRN, it may be because it's a bad bicluster, but also because information about TRN consecutiveness or GO annotations are not complete. Just as an example, E. Coli TRN grew from 424 genes and 577 interactions in 2002 to 1278 genes and 2724 interactions in 2004 . Also statistical significance tests are controversial.

3.2 Non-biological external indices

Non-biological external indices are used to check if bicluster results match with previous knowledge of biclusters in the data. They also can be used in comparing biclusters of two different biclustering methods. There are two main techniques to generate external indices: two-matrix and single-matrix techniques. In case of two-matrix technique, two binary matrices are built, *P* and *R*, of size $n \times n$, where *n* is the number of objects (genes or conditions) of our data. *P* represents the grouping of objects in the a priori partition and *R* the grouping in our results. Frow those two matrices, indices are defined. Though the adaptation of two-matrix technique to bi-dimensionality is not very difficult, the concept of overlapping is harder to express with this method, so single matrix is preferred. Single-matrix technique builds a unique bicluster matrix *M* of order $p \times r$ where *p* is the number of biclusters in *P* and *r* is the number of biclusters in *R*. *mi j* will determine the similarity between the bicluster *i* of *P* and the bicluster *j* of *R*. A measure of this similarity is *F*1 index proposed by Getz et al. and adapted to biclusters by Turner et al.. *F*1 is based in the proportion of bicluster *i* present in bicluster *j* (sensitivity or module recovery of bicluster *i*) and the proportion of bicluster *j* present in bicluster *i* (specificity or relevance of bicluster *i*). Note that the sensitivity of bicluster *i* for *j* is the specificity of bicluster *j* for *i*, and the same with the specificity of *i* for *j*, that is the sensitivity of *j* for *i*. If *gx* is the number of genes in *X*, *cx* the number of conditions in *X* and $nx = gxcx$; sensitivity, specificity and *F*1 are defined as:

$$sensitivity = \frac{(g_{A\cap B})(c_{A\cap B})}{n_B} \quad (14)$$

$$specificity = \frac{(g_{A\cap B})(c_{A\cap B})}{n_A} \quad (15)$$

$$F_1(A,B) = \frac{2(g_{A\cap B})(c_{A\cap B})}{n_A + n_B} \quad (16)$$

When results in *R* reveal exactly a priori partition *P*, *M* will be (if computed with Eq. 16) a square (*p*× *p*), symmetric matrix with *mi j* = 1 if *i* = *j* and *mi j* = *mji* < 1 otherwise. From *M* we can get two measures of the overall matching between *R* and *P*.

$$S(R,P) = \frac{1}{r}\sum_{i=1}^{r} max_{j=1}^{p}(m_{ij}) \quad (17)$$

$$S(P,R) = \frac{1}{p}\sum_{j=1}^{p} max_{i=1}^{r}(m_{ij}) \quad (18)$$

S(*R*,*P*) gives overall bicluster relevance of biclustering *R*, while *S*(*R*,*P*) gives the module recovery capacity of biclustering *R*.

3.3 Internal indices

Internal indices compare intrinsic information about data with the biclustering results. In this case, no a priori information further than the raw data is available. Internal indices are not as precise as external indices, but they are important when a priori information is not available. To avoid the use of internal indices, synthetic data with known structure are built to validate biclustering methods.When applied to real biological data where no a priori information is known, biological tests are used. An internal index is computed from two matrices just as non-biological external indices. In this case, matrix *P* contains information about proximity between expression levels of genes or conditions. Now, *Pi j* = *Pji* = *distance*(*oi*,*oj*). Again two pairs of matrices are needed for biclustering, one where *oi* are genes and another for conditions. *Pi j* is greater when *oi* and *oj* are different. *R* can be built as described for external indices, but inversed so higher values correspond to objects not grouped together. For example*Ci j* =1/(1+*k*), where *k* is the number of times that objects *i* and *j* are grouped together. *Ci j* will be in (0,1], being 1 if never grouped together and downing to near 0 if usually grouped. This two matrices can be compared with normalized Hubert statistic:

$$\bar{\Gamma}(C,P) = \frac{\frac{1}{m}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(P_{ij}-\mu_p)(C_{ij}-\mu_c)}{\sigma_p\sigma_c} \quad (19)$$

For example, Jain and Dubes survey different drawbacks of cophenetic coefficient, estimating than even a value of 0.9 will not be enough to assert that there is a good correlation between *P* and *R*.

3.4 Relative indices

Relative indices try to determine the best choice of our algorithm parameters on each particular data set. If we want to compare two algorithms against the same data set, we want to compare its best parameterization for this data set. However this is a difficult task because of the heterogeneity of the biclustering algorithms and its input parameters. Relative indices use to be external or internal indices, depending on the availability of a priori information from the data. Independently of the index, the procedure is to run the algorithm with different parameter configurations, and compute the index for each one. The parameter configuration with best index is selected as optimal for the data set. Selection of the different parameter configurations is up to the user and is key for the optimal search, so it must represent all the range of possibilities, avoiding deviations. In clustering, another approach to find the best configuration is to find an stable number of clusters, retrieved by a great number of configurations. From them, we take the

one in the middle of the range, or the one with the best value for a given index. This method is also used in some biclustering validations, usually to find stability when the algorithm has pseudo-random behaviour but not to find optimal initial parameters. These indices are used for measuring the quality of a clustering result comparing to other ones which were created by other clustering algorithms, or by the same algorithms but using different parameter values. These indices are usually suitable for measuring crisp clustering. Crisp clustering means having non overlapping partitions.

Table 3.Results of Application of indices on dataset 1

Indices	Row partition	Column partition	Best couple
Dunn	**(3,3)** (3,4) (3,5) (3,6)	**(3,3)** (4,3) (5,3) (6,3)	**(3,3)**
BH	**(3,3)** (3,5) (3,6)	**(3,3)** (5,3) (6,3)	**(3,3)**
HL	(6,2)	(4,6)	x
KL	(4,2)	(3,2)	x
DB	**(3,3)** (3,5) (3,6)	**(3,3)** (4,3) (6,3)	**(3,3)**
CH	**(3,3)** (3,4) (3,5)	**(3,3)** (4,3) (6,3)	**(3,3)**
Silhouette	**(3,3)**	**(3,3)**	**(3,3)**

4 APPLICATIONS

4.1 CROKI2 Algorithm

CROKI2 algorithm is an adapted version of k-means based on the Chi-square distance. It is applied to contingency tables to identify a row partition P and a column partition Q that maximises $\chi 2$ value of the new matrix obtained by grouping rows and columns. CROKI2 consists in applying K-means algorithm on rows and on columns alternatively to construct a series of couples of partitions (Pn,Qn) that optimizes $\chi 2$ value of the new data matrix. Given a contingency table $A(X, Y)$, with set of rows X and set of columns Y, the aim of CROKI2 algorithm is to find a row partition $P = (P1, ..., PK)$ composed of K clusters and a column partition $Q = (Q1, ...,QL)$ composed of L clusters that maximizes $\chi 2$ value of the new contingency table (P, Q) obtained by grouping rows and columns in respectively K and L clusters. The criterion optimized by the algorithm is :

$$\chi^2(P,Q) = \sum_{k=1}^{K}\sum_{l=1}^{L} \frac{(f_{kl} - f_{k.}f_{.l})^2}{f_{k.}f_{.l}}$$

with

$$f_{kl} = \sum_{i\in P_k}\sum_{j\in Q_l} f_{ij}$$

$$f_{k.} = \sum_{l=1,L} f_{kl} = \sum_{i\in P_k} f_{i.}$$

$$f_{.l} = \sum_{k=1,K} f_{kl} = \sum_{j\in Q_l} f_{.j}$$

The new contingency table $T_1(P,Q)$ is defined by this expression:

$$T_1(k,l) = \sum_{i\in P_k}\sum_{j\in Q_l} a_{ij}$$

The new contingency table $T1(P,Q)$ is defined by this expression:

$$T_1(k,l) = \sum_{i\in P_k}\sum_{j\in Q_l} a_{ij}$$

We have applied some of the performance measures discussed to two biclustering algorithms, Bimax and improved Plaid Model of Turner et al Bimax is one of the most compared biclustering methods, by means of non-biological and biological validation. For example, in , non-biological measures are used, but only based in gene dimension because hierarchical clustering was one of the methods compared. Also, in the mentioned comparison only default parameters are used for each algorithm, no parameter optimization is done. Turner plaid model was tested by their authors with different synthetic data sets with three to ten (overlapped in different proportions) biclusters. Turner and Bimax algorithms have never been compared in bibliography. Both methods have been implemented in R according to the specifications in the corresponding bibliography. Bimax density of 1s against 0s is proved in a range from 1% to 10% (steps of 1%). Turner's *t*1 and *t*2 parameters are proved as *t*1 = *t*2 in a range from 0.4 to 0.8, with steps of 0.1.

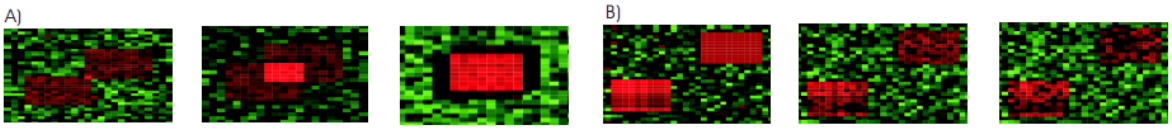

Fig. 2. A) Overlapped constant overexpression biclusters. A low noise has been added to biclusters. Overlapping degree is the same in rows and columns. B) Constant and coherent overexpression biclusters with random noise. Note how noise affects the structure of biclusters, being constancy undistinguishable from coherency with high noise.

About constancy and coherence measures (Fig. 3 b-2), the measures increase with noise, revealing how structure is eventually lost. Additive coherent bicluster has lower (better) *H+* measure than *C* measure, as expected. Note how *H+* measures increase with noise until, eventually, surpassing *C* measure and coinciding with Bimax performance downgrade.

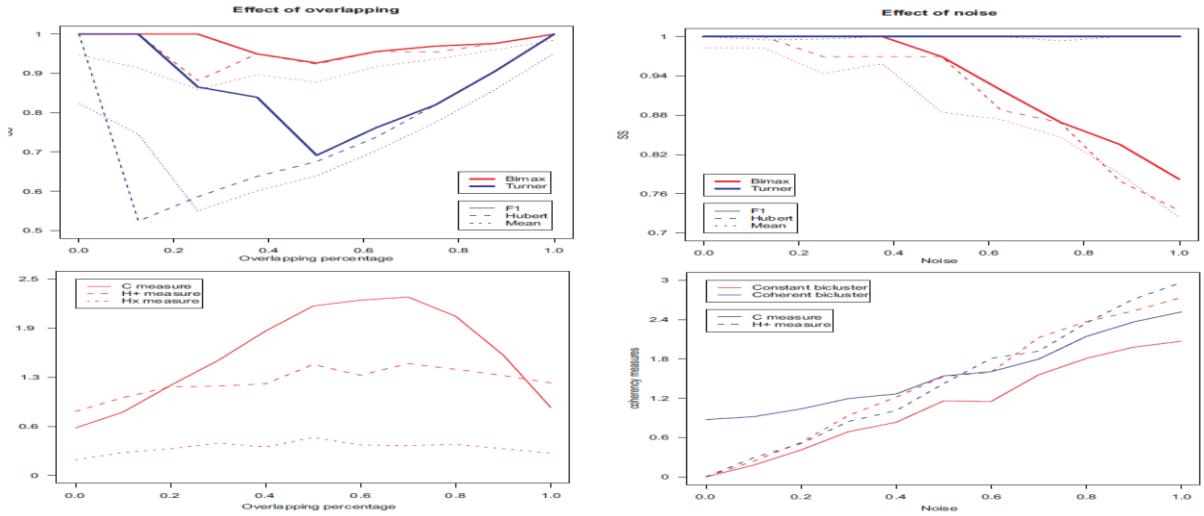

Fig. 3. a) Effect of overlapping in the algorithm and the biclusters. 1) Best *SS* measure achieved by using *F*1 statistics along with the mean of *SS* for all the proven configurations. 2) Variation in the measures of constancy and coherency with changes in the overlap degree. b1) and b2) As a1) and a2), but representing the effect of the noise in the algorithms and biclusters, respectively.

5. Conclusions and Future Work

To extend the use of some indices used initially for classic clustering to biclustering algorithms, especially CROKI2 algorithm for contingency tables generating a framework that will define bicluster specific measures (relative, internal and external indices), data type definitions (constant, coherent), benchmark algorithms and example (real and synthetic) data sets. Though external indices use is extended, our approach to relative and internal index application is new. That helps in automatic optimization of biclustering input parameters, a task seldom considered and critical for obtaining the highest performance. Data type definition exists as discussed, but only constant biclusters have been mathematically measured. Those indices are able to find correct number of clusters when applied to data sets

with diagonal structure i.e data sets having the same number of clusters on rows and columns We present an approach to measure coherence biclusters by using constant measures and transformation of data matrices.

References

[1] Charrad, M.: une approche generique pour l'analyse croisant usage et contenu de sites Web par des methodes de bipartitionnement. PhD Thesis, Paris (2010)

[2] Cheng, Y., Church, G.M.: Biclustering of expression data. In: 8th Int. Conf. On Intelligent Systems for Molecular Biology, pp. 93–103 (2000)

[3] Costa, G., Manco, G., Ortale, R.: A hierarchical model-based approach to coclustering high-dimensional data. In: ACM sym. on App. comput., pp. 886–890 (2008)

[4] Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: 7th ACM SIGKDD 2001, California, pp. 269–274 (2001)

[5] Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: ACM SIGKDD, pp. 89–98. ACM, Washington DC (2003)

[6] Getoor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of link structure. J. Mach. Learn. Res. 3, 679–707 (2002)

[7] A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988.

[8] M. Kanehisa and S. Goto. Kegg: Kyoto encyclopedia of genes and genomes. Nucleic Acids Research, 28(1):27–30, 2000.

[9] H.-W. Ma, B. Kumar, U. Ditges, F. Gunzer, J. Buer, and A.-P. Zeng. An extended transcriptional regulatory network of escherichia coli and analysis of its hierarchical structure and network motifs. Nucleic Acids Research, 32(22):6643–6649, 2004.

[10] S. Madeira and A. Oliveira. Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions of Computational Biology and Bioinformatics, 1(1):24–45, 2004.

[11] R.Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. Science, 298:824–827, 2002.

[12] A. Prelic, S. Bleuer, P. Zimmermann, A. Wille, P. Bhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics, 22(9):1122–1129, 2006.

[13] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. Nature Genetics, 31:64–68, 2002.

[14] H. Turner, T. Bailey, and W. Krzanowski. Improved biclustering of microarray data demonstrated through systematic performance tests. Computational Statistics and Data Analysis, 48:235–254, 2003.

[15] H. L. Turner, T. C. Bailey, W. J. Krzanowski, and C. A. Hemingway. Biclustering models for structured microarray data. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2(4):316–329, 2005.