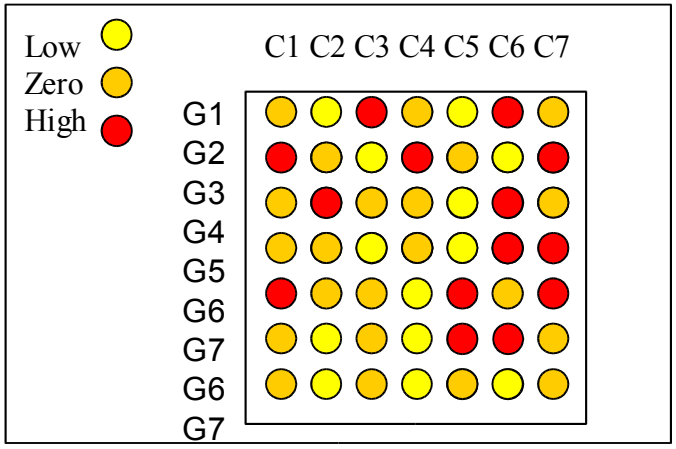
BICLUSTERING OF GENE EXPRESSION DATA

Jesús S. Aguilar-Ruiz
Computer Science Department
University of Seville, SPAIN

http://www.lsi.us.es/~aguilar
aguilar@lsi.us.es

Microarrays: Gene Expression Data

- Microarray analysis allows the monitoring of the activities of many genes over many different conditions.
- Experiments are carried out on a Physical Matrix like the one below:



Conditions →

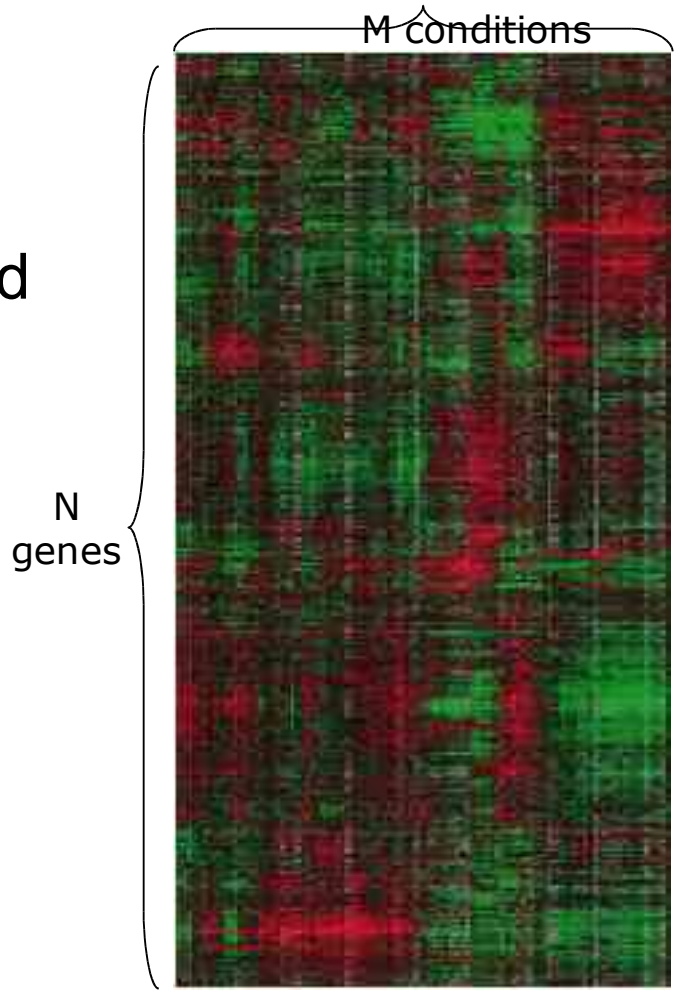
Genes ↓

1.55	1.05	0.5	2.5	1.75	0.25	0.1
1.7	0.3	2.4	2.9	1.5	0.5	1.0
1.55	1.05	0.5	2.5	1.75	0.25	0.1
1.7	0.3	2.4		1.5	0.5	1.0
1.55		0.5	2.5	1.75	0.25	0.1
	0.3	2.4	2.9	1.5	0.5	1.0
1.55	1.05	0.5	2.5	1.75	0.25	0.1

To facilitate computational analysis the physical matrix which may contain 1000's of gene's is converted into a numerical matrix using image analysis equipment.

Microarrays: Gene Expression Data

- It is common to visualize a gene expression datasets by a color plot:
  - Red spots: high expression values (the genes have produced many copies of the mRNA).
  - Green spots: low expression values.
  - Gray spots: missing values.



Microarrays: Gene Expression Data

Microarray data can be viewed as an N×M matrix:

- Each of the N rows represents a gene (clone, ORF, etc.).
- Each of the M columns represents a condition (a sample, a time point, etc.).
- Each entry represents the expression level of a gene under a condition. It can either be an absolute value (e.g. Affymetrix GeneChip) or a relative expression ratio (e.g. cDNA microarrays).
- A row/column is sometimes referred to as the "expression profile" of the gene/condition.

Clustering

- Cluster Analysis is an unsupervised procedure which involves grouping of objects based on their similarity in feature space.
- In the Gene Expression context **Genes** are grouped based on the similarity of their **Condition** feature profile.

Number of ways in which **n** examples can be partitioned into **k** non-empty subsets:

$$P(n,k) = \frac{1}{k!}\sum_{j=0}^{k}\binom{k}{j}(-1)^j(k-j)^n$$

An approximation:

$$P(n,k) \approx \frac{k^n}{k!} \approx k^{n-k}e^k\sqrt{2\pi k}$$

If we do not know the number of clusters **k**, the total number of evaluations is:

$$T(n) = \sum_{k=1}^{n} P(n,k)$$

For example,

$$n=8,\ T(8) = 4140$$

Clustering

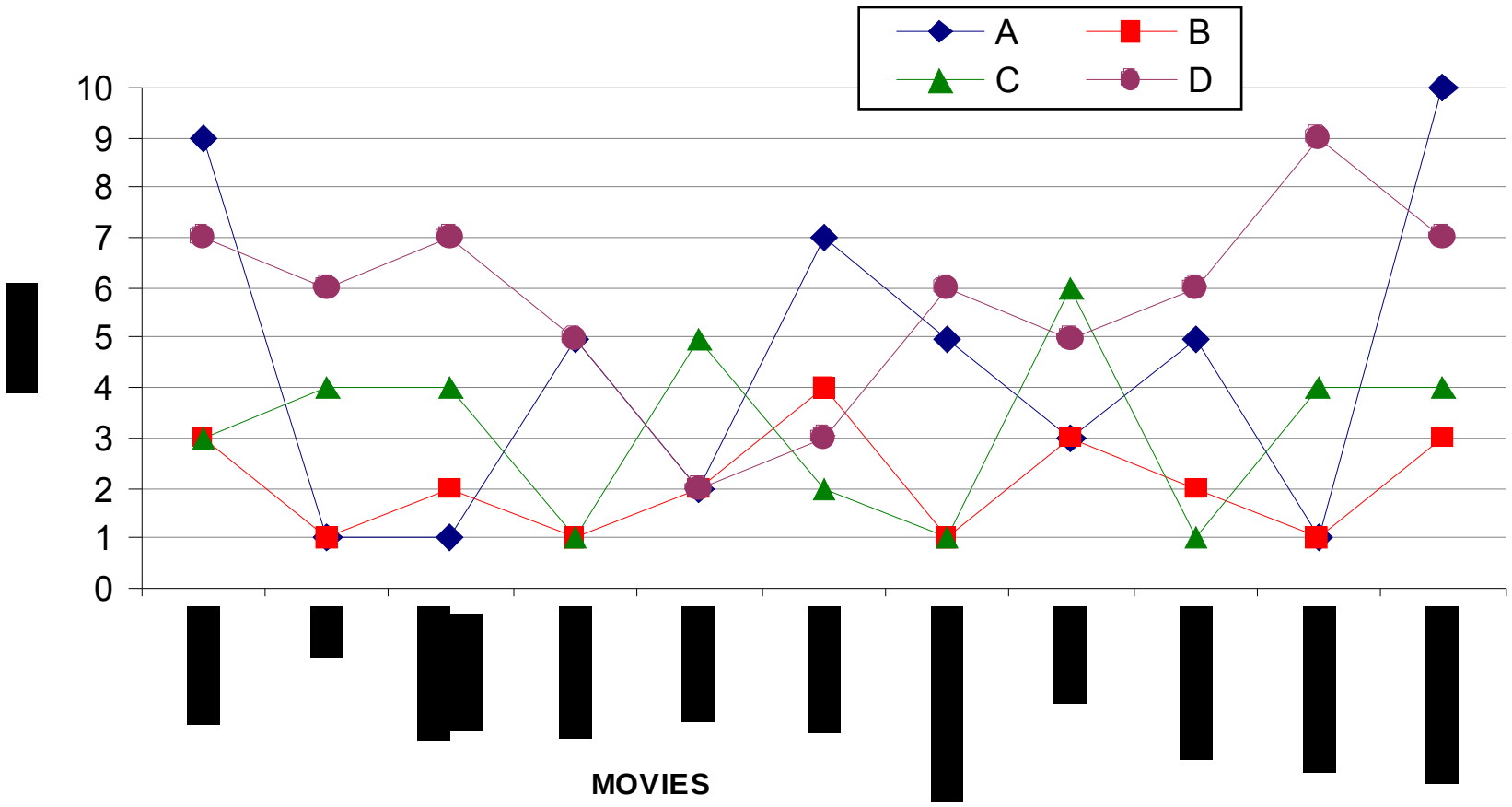
- Requirements:
  - Parameter: number of clusters
  - Distance
- Distance is defined for the M-dimensional space.
  - What distance metric?
  - Normalized/Standarized?
  - Any outlier?
  - Missing values?

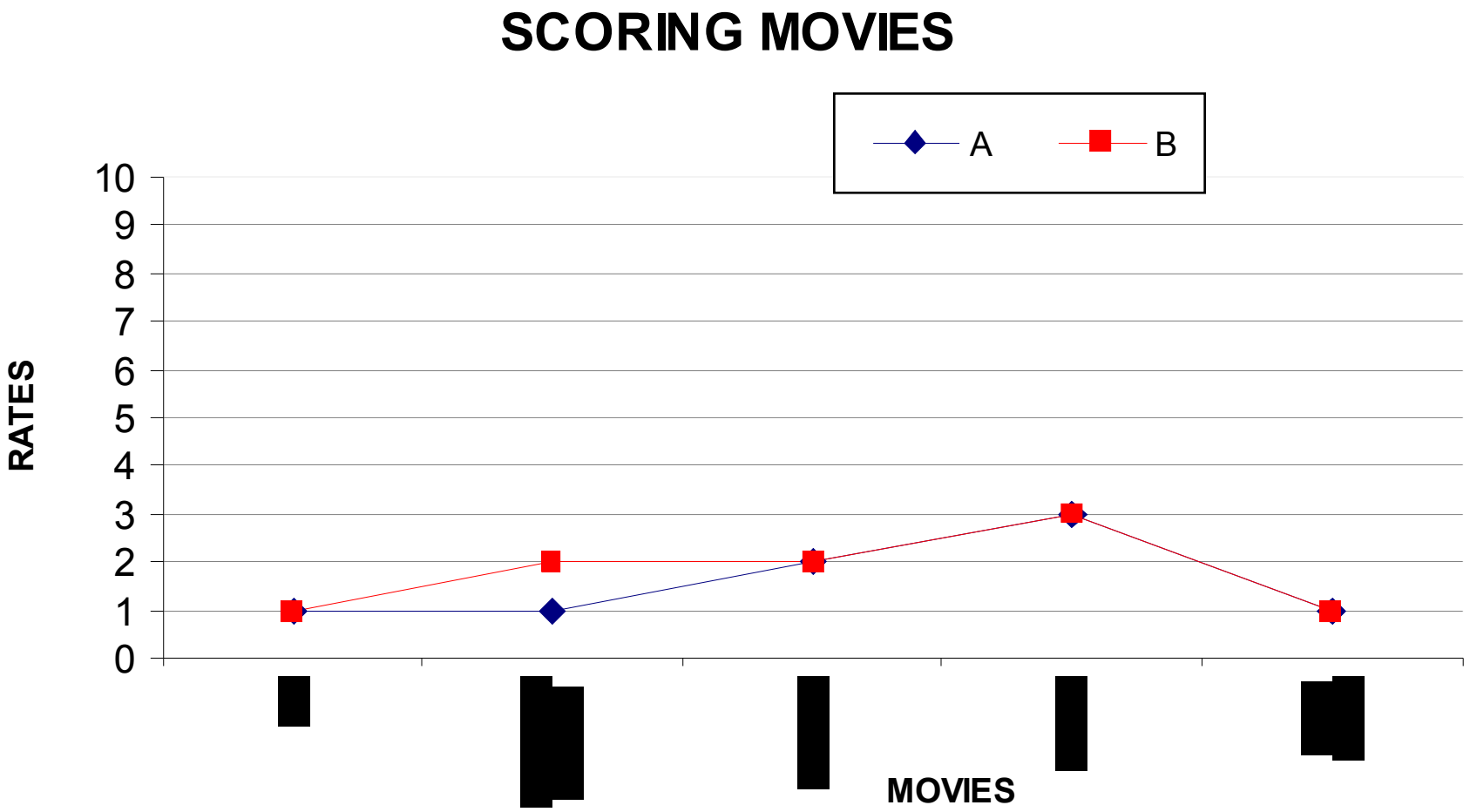
Example I

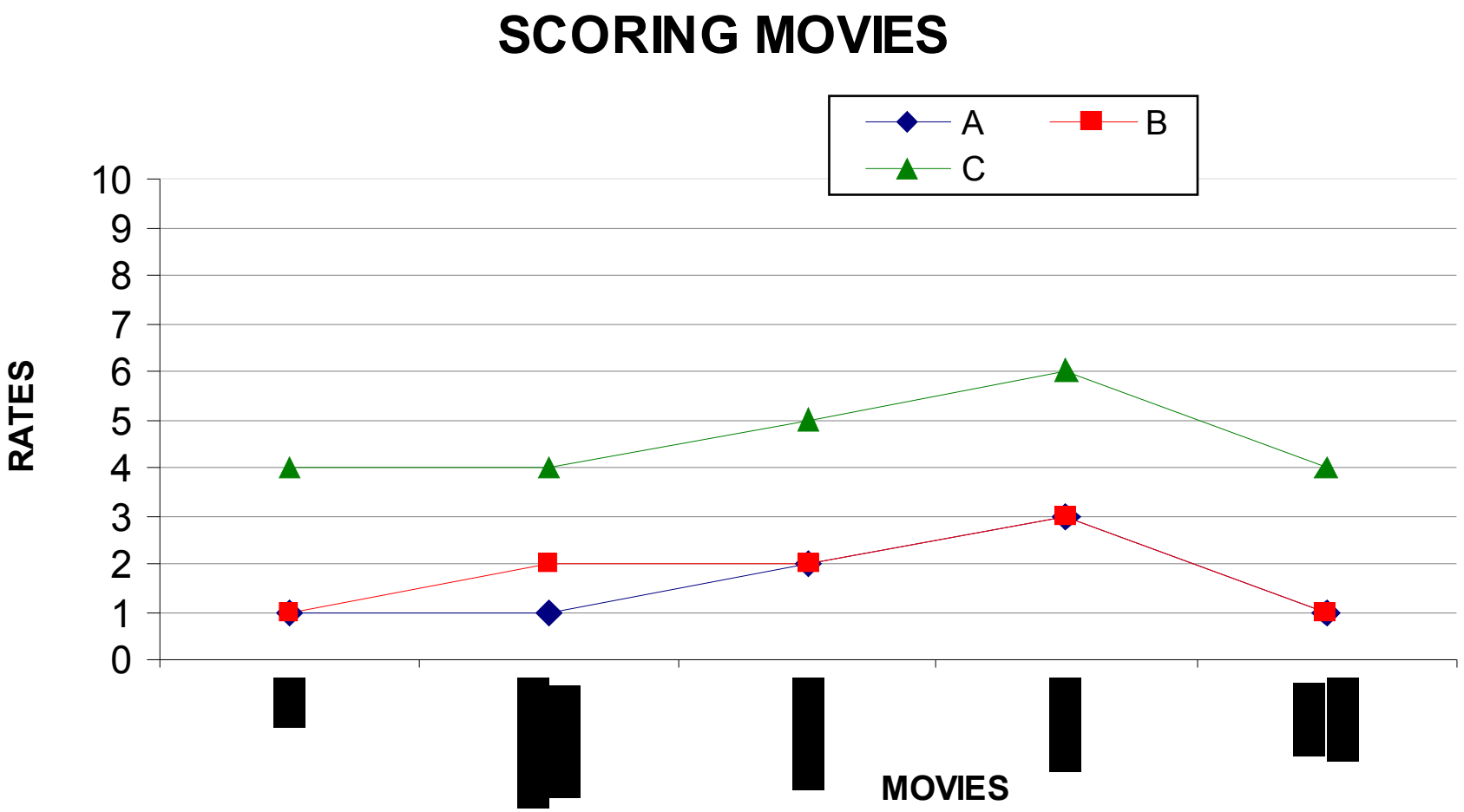
SCORING

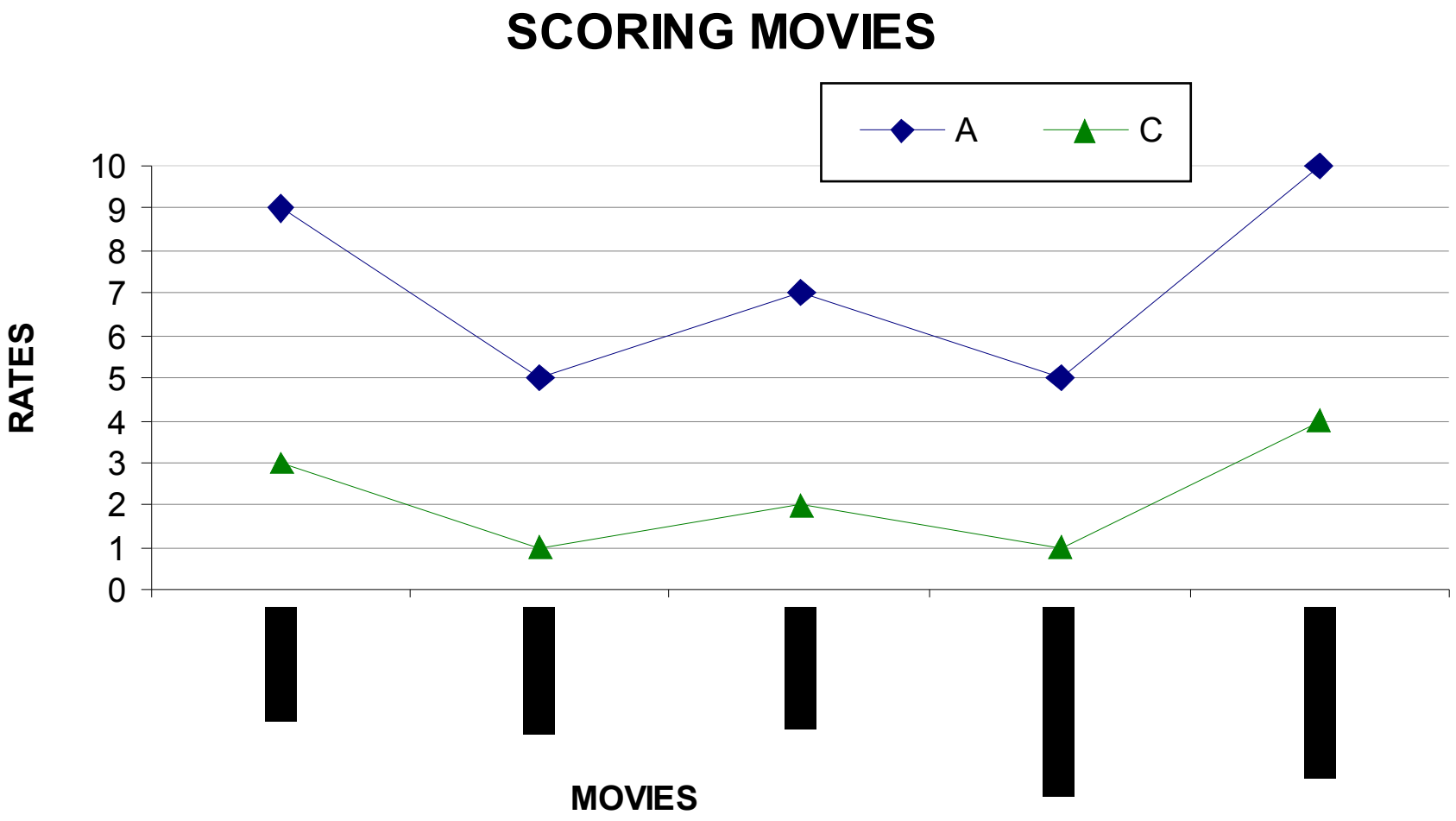
MOVIE	A	[illegible]	[illegible]	[illegible]
MINORITY REPORT	8	[illegible]	[illegible]	[illegible]
SHREK 2	9	[illegible]	[illegible]	[illegible]
JFK	1	[illegible]	[illegible]	[illegible]
RT	1	[illegible]	[illegible]	[illegible]
RAIN MAN	5	[illegible]	[illegible]	[illegible]
PATTON	2	[illegible]	[illegible]	[illegible]
PHILADELPHIA	7	[illegible]	[illegible]	[illegible]
CHICAGO	5	[illegible]	[illegible]	[illegible]
TORRENTE	3	[illegible]	[illegible]	[illegible]
STAR WARS	9	[illegible]	[illegible]	[illegible]
$ FOR Y	1	[illegible]	[illegible]	[illegible]

SCORING MOVIES

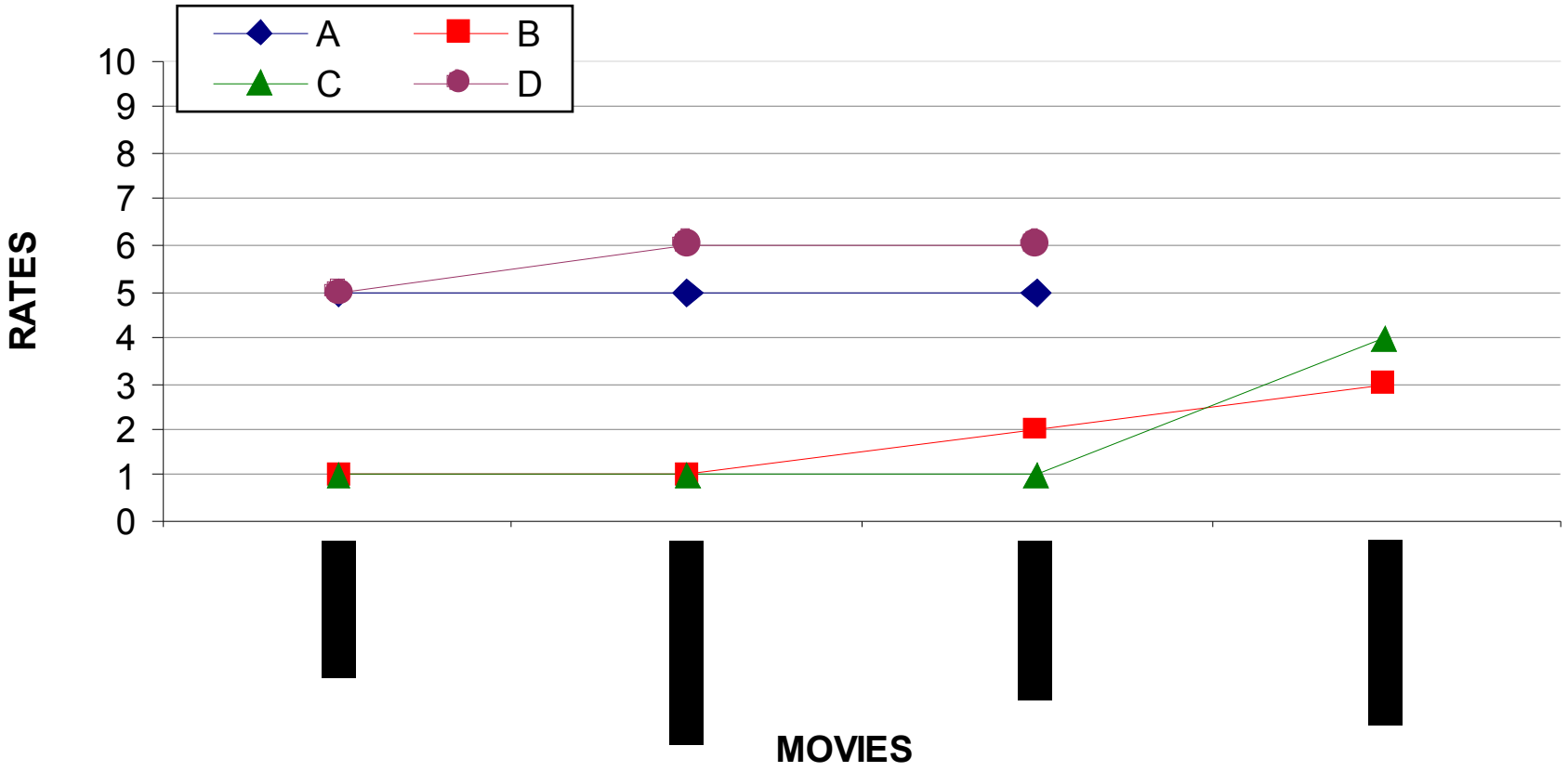



SCORING MOVIES
A
B
10
9
8
7
6
5
4
3
2
1
0
RATES
MOVIES


SCORING MOVIES
A
B
C
10
9
8
7
6
5
4
3
2
1
0
RATES
MOVIES


SCORING MOVIES
A
C
10
9
8
7
6
5
4
3
2
1
0
RATES
MOVIES

SCORING MOVIES



BICLUSTERING

- Grouping examples for a subset of attributes.

SUBSPACE CLUSTERING

- Types:
  - Constant patterns
  - Similar patterns
  - Shifting patterns
  - Scaling patterns

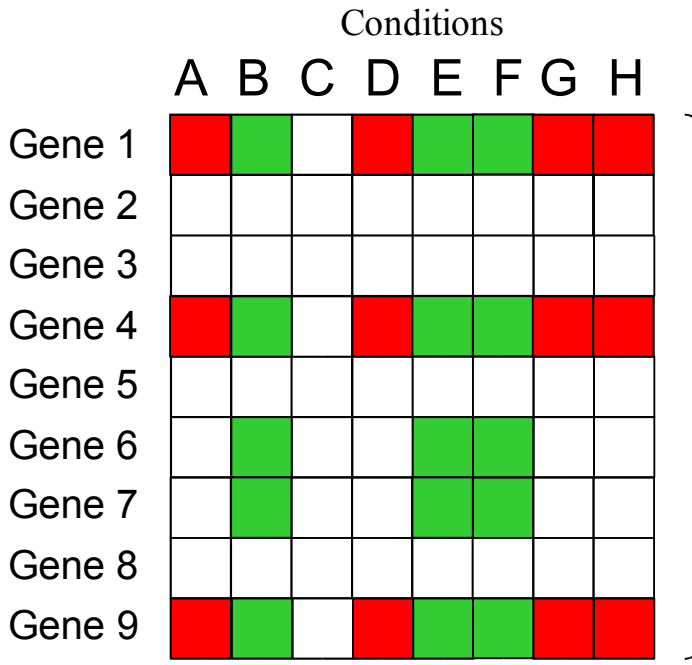
- Properties:
  - $B_i \cap B_j \neq \phi$
  - $\bigcup_i B_i \neq S$

- Challenges:
  - Curse of dimensionality
  - Similarity Model needed

- Complexity: $O(k2^{n+m})$

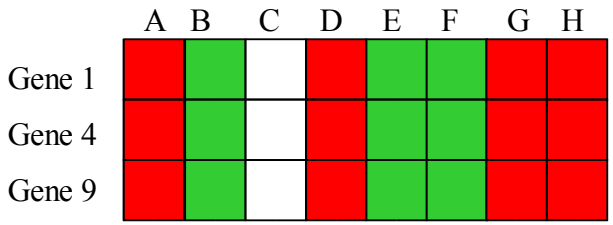
Biclustering

Technique first described by J.A. Hartigan in 1972 and termed 'Direct Clustering'.



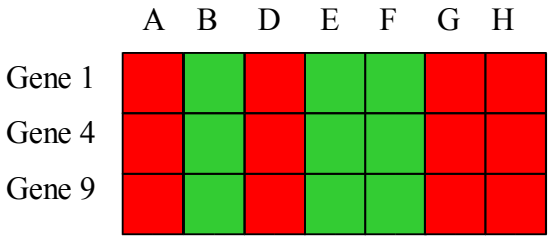
First introduced to Microarray expression data by Cheng and Church(2000)

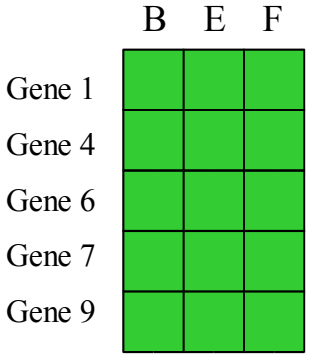
Clustering



Biclustering

Biclustering discovers local coherences over a subset of conditions





Types of biclusters

- Biclusters with constant values
- Biclusters with constant values on rows or columns
- Biclusters with coherent values
- Biclusters with coherent evolutions

Biclusters with constant values

- Biclusters with constant values
- Biclusters with constant values on rows or columns

1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0

a) Constant Bicluster

1.0	1.0	1.0	0.0
2.0	2.0	2.0	2.0
3.0	3.0	3.0	3.0
4.0	4.0	4.0	4.0

b) Constant Rows

1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0

c) Constant Columns

- A perfect constant bicluster is a submatrix (I,J), where all values within the bicluster are equal for all i∈I and all j ∈J.
- A perfect bicluster with constant rows is a sub-matrix (I,J), where all the values within the bicluster can be obtained using one of the following expressions:
- A perfect bicluster with constant columns is a sub-matrix (I,J), where all the values within the bicluster can be obtained using one of the following expressions:

$$a_{ij} = \mu$$

$$a_{ij} = \mu + \alpha_i$$

$$a_{ij} = \mu \times \alpha_i$$

$$a_{ij} = \mu + \beta_j$$

$$a_{ij} = \mu \times \beta_j$$

Biclusters with coherent values

1.0	2.0	5.0	0.0
2.0	3.0	6.0	1.0
4.0	5.0	8.0	3.0
5.0	6.0	9.0	4.0

d) Coherent Values – Additive Model

1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

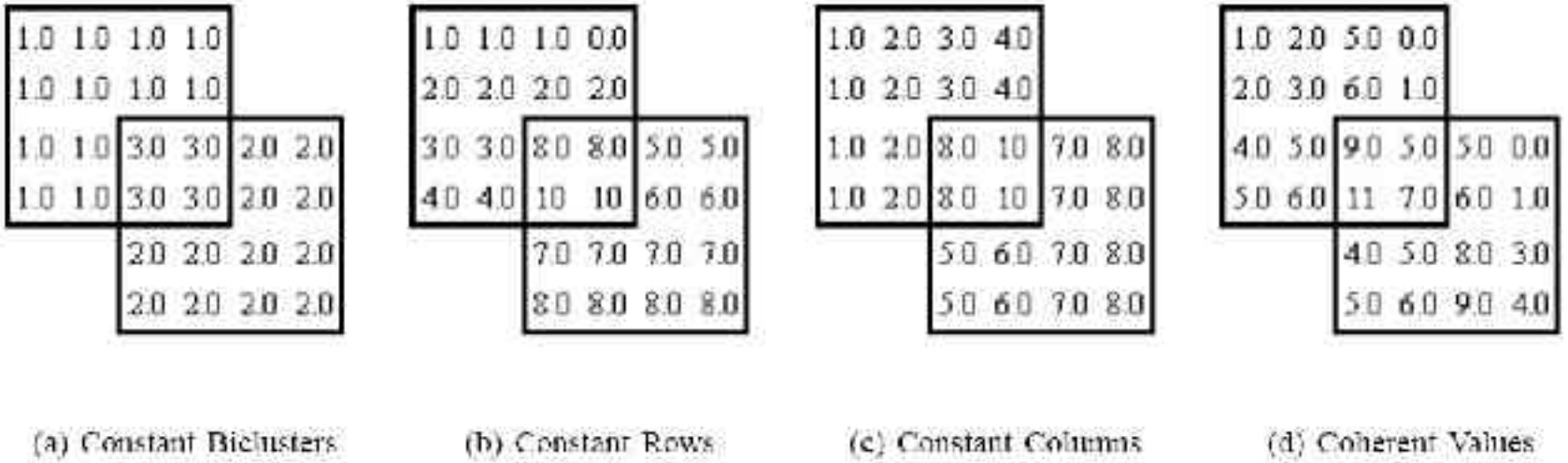
e) Coherent Values – Multiplicative Model

A perfect bicluster with coherent values, is defined as a subset of rows and a subset of columns, whose values are predicted using the following expression:

- Additive Model : $a_{ij} = \mu + \alpha_i + \beta_j$
- Multiplicative Model: $a_{ij} = \mu' \times \alpha'_i \times \beta'_j$

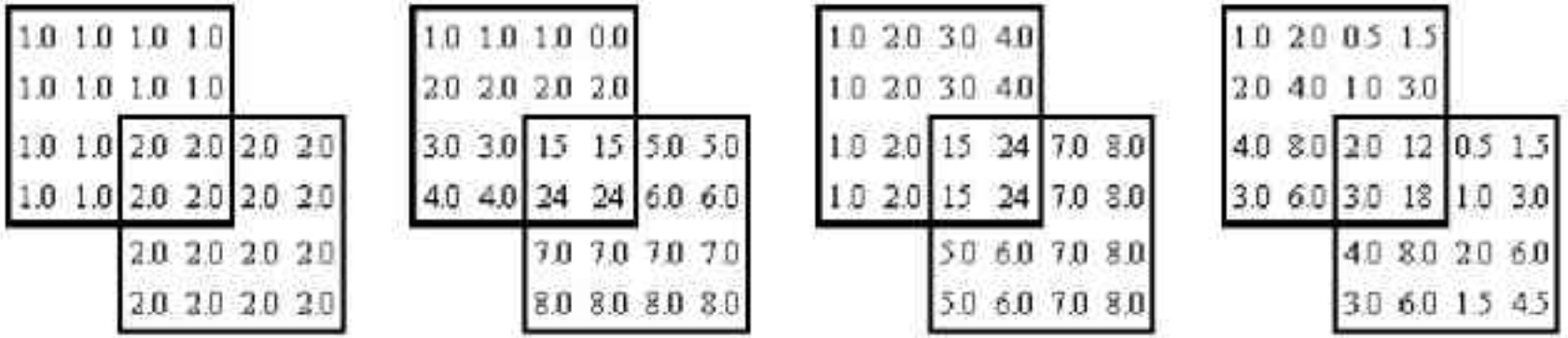
Overlapping

General Additive Model



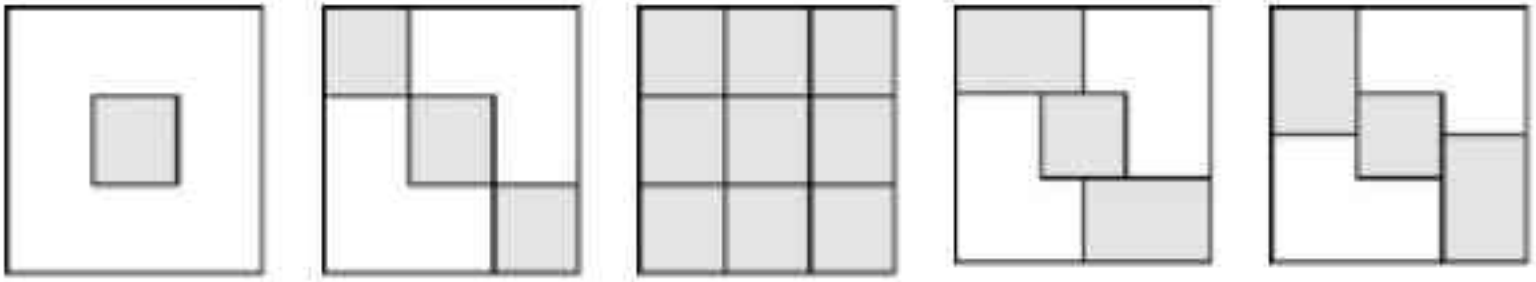
(a) Constant Biclusters (b) Constant Rows (c) Constant Columns (d) Coherent Values

General Multiplicative Model



(a) Constant Biclusters (b) Constant Rows (c) Constant Columns (d) Coherent Values

Structure



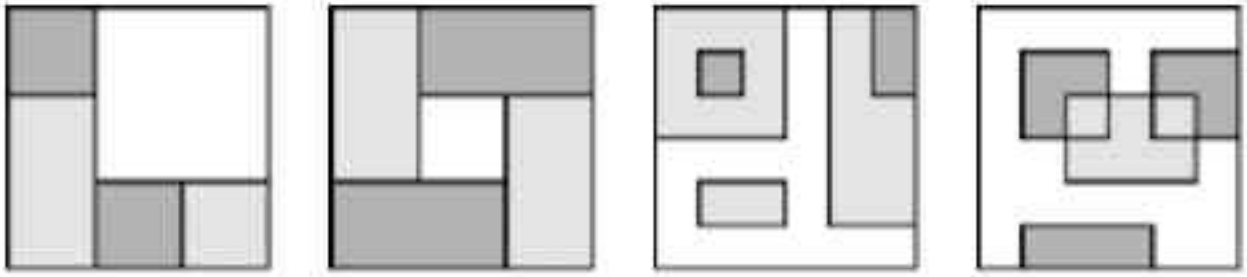
(a) Single Bicluster

(b) Exclusive row and column biclusters

(c) Checkerboard structure

(d) Exclusive-rows biclusters

(e) Exclusive-columns biclusters



(f) Non-Overlapping biclusters with tree structure

(g) Non-Overlapping non-exclusive biclusters

(h) Overlapping biclusters with hierarchical structure

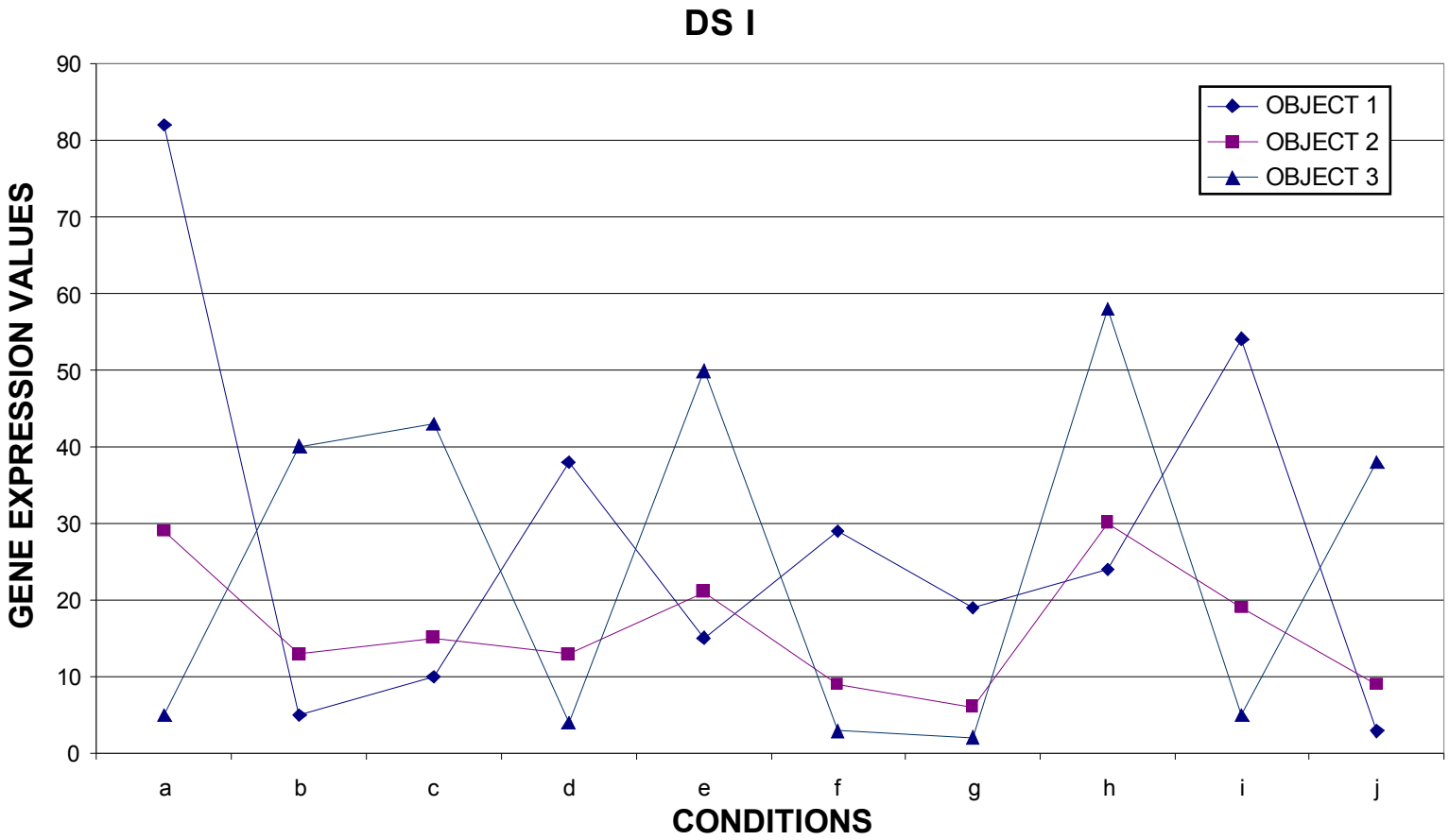
(i) Arbitrarily positioned overlapping biclusters

Biclusters with coherent evolutions

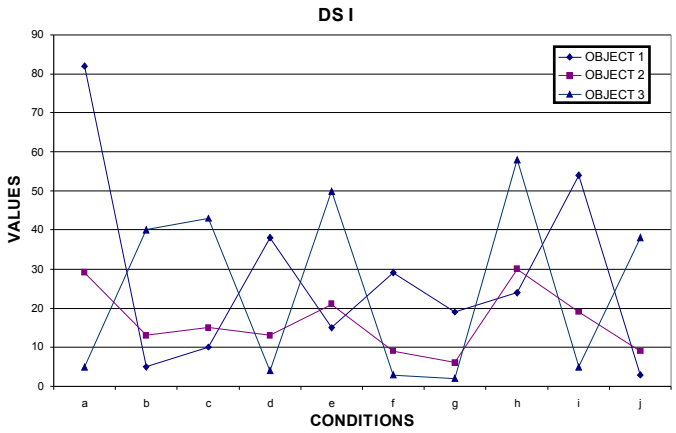
- Evidence that a subset of genes is up-regulated or down-regulated across a subset of conditions without taking into account their actual expression values.
- Order-preserving sub-matrix (OPSM)

70	13	19	10
92	40	49	35
40	20	27	15
90	15	20	12

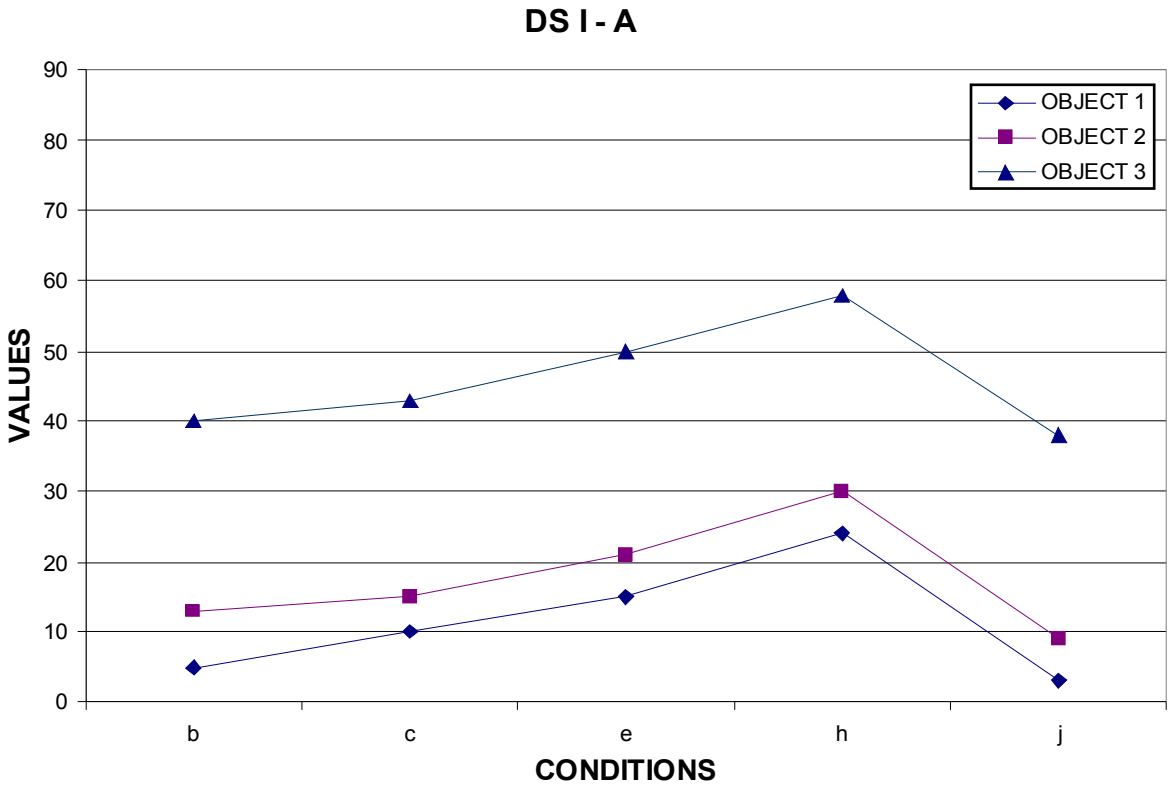
Patterns


DS I
GENE EXPRESSION VALUES
CONDITIONS
OBJECT 1
OBJECT 2
OBJECT 3
0
10
20
30
40
50
60
70
80
90
a
b
c
d
e
f
g
h
i
j

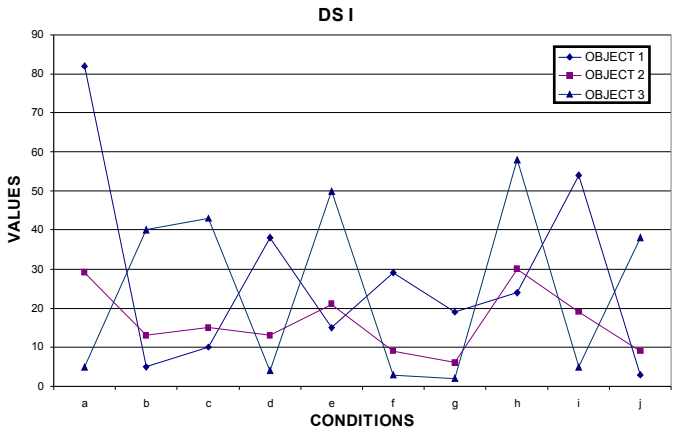
Shifting Pattern



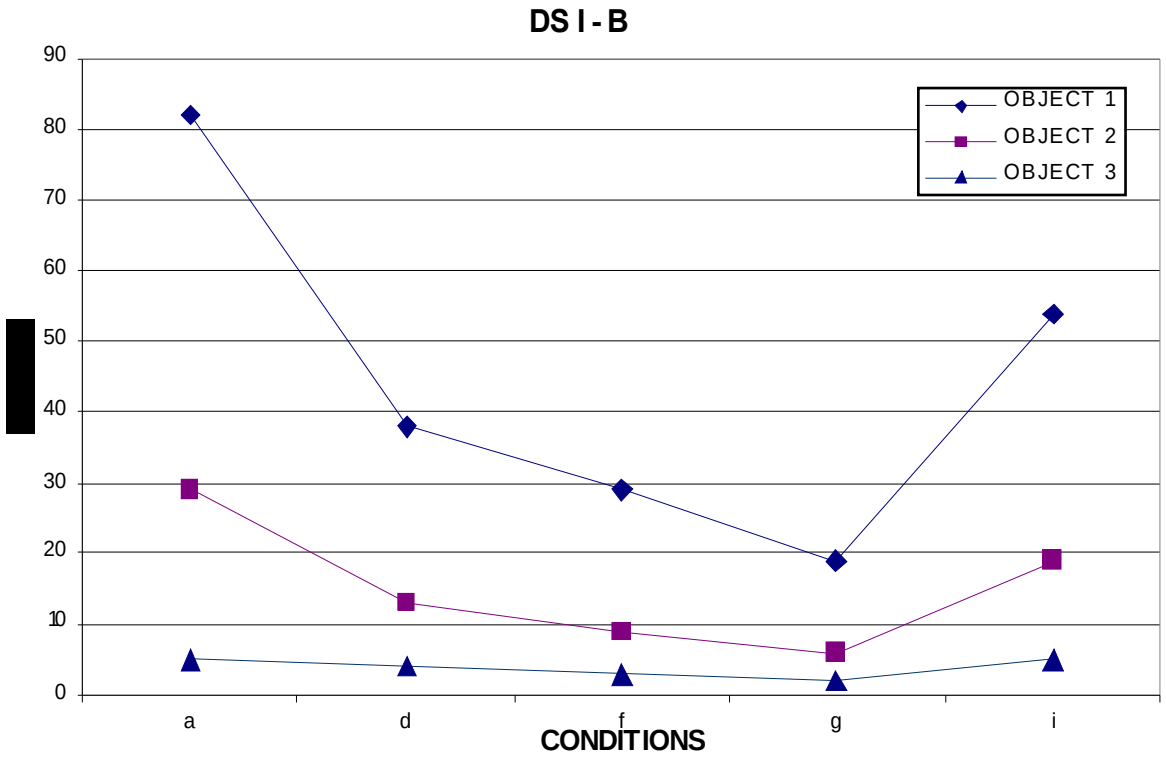
Shifting Pattern in subspace {b,c,e,h,j}



Scaling Pattern



Scaling Pattern in subspace {a,d,f,g,i}



Overview of Methods

Method	Publish	Allow overlap?	Complexity	Testing Data
Cheng & Church	ISMB 2000	Yes (rare in reality)	O(MN) or O (MlogN)	Yeast (2884×17), lymphoma (4026×96)
Getz et al. (CTWC)	PNAS 2000	Yes	Exponential	Leukemia (1753×72), colon cancer (2000×62)
Lazzeroni & Owen (Plaid Models)	Bioinformatics 2000	Yes	Polynomial	Food (961×6), forex (276×18), yeast (2467×79)
Ben-Dor et al. (OPSM)	RECOMB 2002	Yes	$O(NM^3l)$	Breast tumor (3226×22)
Tanay et al. (SAMBA)	Bioinformatics 2002	Yes	$O((N2^{d+1})^{\log_{(r+1)/r(rd)}})$	Lymphoma (4026×96), yeast (6200×515)
Yang et al. (FLOC)	BIBE 2003	Yes	$O((N+M)^2kp)$	Yeast (2884×17)
Kluger et al. (Spectral)	Genome Res. 2003	No	Polynomial	Lymphoma (1 rel., 1 abs.), leukemia, breast cell line, CNS embryonal tumor

Cheng & Church's algorithm

- Model:
  - A bicluster is represented by the submatrix A of the whole expression matrix (the involved rows and columns need not be contiguous in the original matrix).
  - Each entry A_{ij} in the bicluster is the superposition (summation) of:
    1. The background level
    2. The row (gene) effect
    3. The column (condition) effect
  - A dataset contains a number of biclusters, which are not necessarily disjoint.

Cheng & Church's algorithm

In the matrix A the residue score of element a_{ij} is given by:

$$R(a_{ij}) = a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}$$

where a_{iJ} = mean of row i,

a_{Ij} = mean of column j,

a_{IJ} = mean of A.

- Biological meaning: the genes have the same (amount of) response to the conditions.

Cheng & Church's algorithm

- Goal: to find biclusters with **minimum squared residue**:

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

This Global *H* score gives an indication of how the data fits together within that matrix- whether it has some coherence or is random.

- A **high** H value signifies that the data is uncorrelated.
- A **low** H score means that there is a correlation in the matrix
  - A score of ***H(I,J)= 0*** would mean that the data in the matrix fluctuates in unison i.e. **the sub-matrix is a bicluster.**

- For an ideal bicluster,
  - H(I, J) = 0.
  - adding a constant to all entries of a row or column yields an ideal bicluster.
  - multiplying all entries in the bicluster by a constant yields an ideal bicluster.

Cheng & Church's algorithm

Matrix (M) Avg. = 6.5

			Row Avg.
1	2	3	2
4	5	6	5
7	8	9	8
10	11	12	11
Col Avg. 5.4	6.4	7.4	

$$R(a_{ij}) = a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}$$

R(1) = 1- 2 - 5.4 + 6.5 = 0.1

R(2) = 2 - 2 - 6.4 + 6.5 = 0.1

⋮ ⋮

R(12) = 12 - 11 -7.4 + 6.5 = 0.1

H (M) = (0.01x12)/12 = 0.01

If 5 was replaced with 3 then the score would changed to: $H(M_2) = 2.06$

If the matrix was reshuffled randomly the score would be around:

$H(M_3) = sqr(12-1)/12 = 10.08$

Cheng & Church's algorithm

- Constraints:
  - $1 \times M$ and $N \times 1$ matrixes always give zero residue.
    => Find biclusters with maximum sizes, with residues not more than a threshold δ (largest δ-biclusters).
  - Constant matrixes always give zero residue.
    => Use average row variance to evaluate the "interestingness" of a bicluster. Biologically, it represents genes that have large change in expression values over different conditions.

Cheng & Church's algorithm

- Finding the largest δ-bicluster:
  - The problem of finding the largest square δ-bicluster ($|I| = |J|$) is NP-hard.
  - Objective function for heuristic methods (to minimize):

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

=> sum of the components from each row and column, which suggests simple greedy algorithms to evaluate each row and column independently.

Cheng & Church's algorithm

Algorithm 0 (Brute-Force Deletion and Addition).

Input: A, a matrix of real numbers, and $\delta \geq 0$, the maximum acceptable mean squared residue score.

Output: A_{IJ}, a δ-bicluster that is a submatrix of A with row set I and column set J, with a score no larger than δ.

Initialization: I and J are initialized to the gene and condition sets in the data and $A_{IJ} = A$.

Iteration:

1. Compute the score H for each possible row/column addition/deletion and choose the action that decreases H the most. If no action will decrease H, or if $H <= \delta$, return A_{IJ}.

A0 → Time complexity: O((N+M)MN)

A1→ Time complexity: O(MN)

Algorithm 1 (Single Node Deletion).

Input: A, a matrix of real numbers, and $\delta \geq 0$, the maximum acceptable mean squared residue score.

Output: A_{IJ}, a δ-bicluster that is a submatrix of A with row set I and column set J, with a score no larger than δ.

Initialization: I and J are initialized to the gene and condition sets in the data and $A_{IJ} = A$.

Iteration:

1. Compute a_{iJ} for all $i \subset I$, a_{Ij} for all $j \subset J$, a_{IJ}, and $H(I, J)$. If $H(I, J) <= \delta$, return A_{IJ}.
2. Find the row $i \in I$ with the largest

$$d(i) = \frac{1}{|J|} \sum_{j \subset J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

and the column $j \subset J$ with the largest

$$d(j) = \frac{1}{|I|} \sum_{i \in I} (a_{ij} \quad a_{iJ} \quad a_{Ij} \; | \; a_{IJ})^2$$

remove the row or column whichever with the larger d value by updating either I or J.

Cheng & Church's algorithm

Algorithm 2 (Multiple Node Deletion).

Input: A, a matrix of real numbers, $\delta \geq 0$, the maximum acceptable mean squared residue score, and $\alpha > 1$, a threshold for multiple node deletion.

Output: A_{IJ}, a δ-bicluster that is a submatrix of A with row set I and column set J, with a score no larger than δ.

Initialization: I and J are initialized to the gene and condition sets in the data and $A_{IJ} = A$.

Iteration:

1. Compute a_{iJ} for all $i \subset I$, a_{Ij} for all $j \subset J$, a_{IJ}, and $H(I, J)$. If $H(I, J) <= \delta$, return A_{IJ}.
2. Remove the rows $i \in I$ with

$$\frac{1}{|J|} \sum_{j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 > \alpha H(I, J)$$

3. Recompute a_{Ij}, a_{IJ}, and $H(I, J)$.
4. Remove the columns $j \subset J$ with

$$\frac{1}{|I|} \sum_{i \in I} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 > \alpha H(I, J)$$

5. If nothing has been removed in the iterate, switch to Algorithm 1.

Algorithm 3 (Node Addition).

Input: A, a matrix of real numbers, I and J signifying a δ-bicluster.

Output: I' and J' such that $I \subset I'$ and $J \subset J'$ with the property that $H(I', J') \leq H(I, J)$.

Iteration:

1. Compute a_{iJ} for all i, a_{Ij} for all j, a_{IJ}, and $H(I, J)$.
2. Add the columns $j \notin J$ with

$$\frac{1}{|I|} \sum_{i \in I} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 < \Pi(I, J)$$

3. Recompute a_{iJ}, a_{IJ}, and $H(I, J)$.
4. Add the rows $i \notin I$ with

$$\frac{1}{|J|} \sum_{j \subset J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 < H(I, J)$$

5. For each row i still not in I, add its inverse if

$$\frac{1}{|J|} \sum_{j \subset J} (-a_{ij} + a_{iJ} - a_{Ij} + a_{IJ})^2 < H(I, J)$$

6. If nothing is added in the iterate, return the final I and J as I' and J'.

Cheng & Church's algorithm

Algorithm 4 (Finding a Given Number of Biclusters).

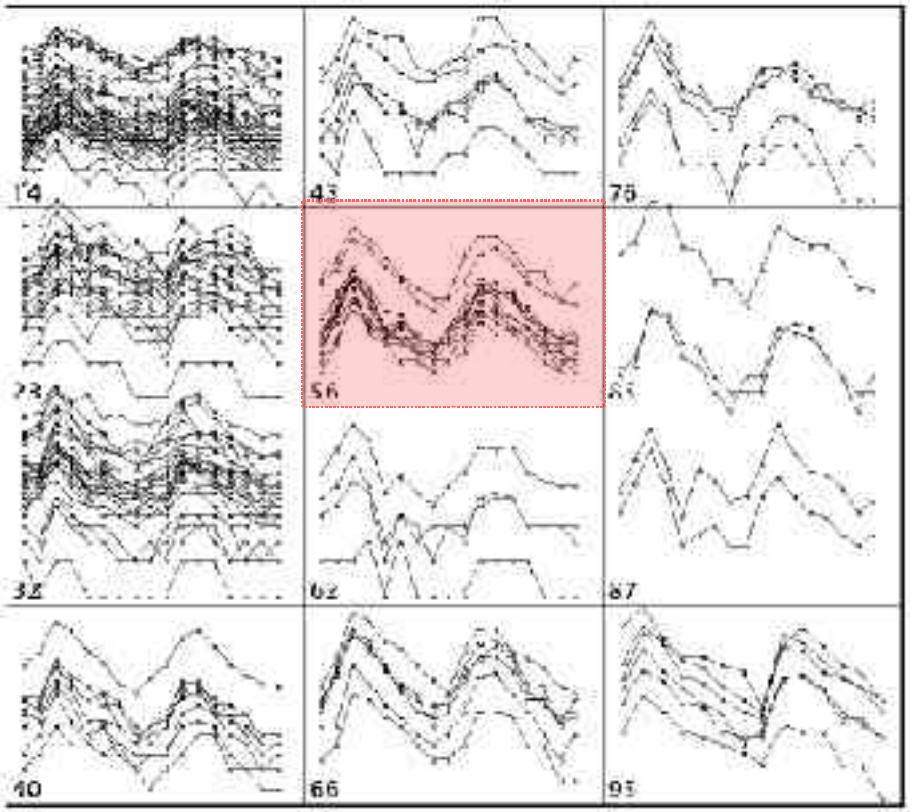
Input: A, a matrix of real numbers with possible missing elements, $\alpha > 1$, a parameter for multiple node deletion, $\delta > 0$, the maximum acceptable mean squared residue score, and n, the number of δ-biclusters to be found.

Output: n δ-biclusters in A.

Initialization: Missing elements in A are replaced with random numbers from a range covering the range of non-null values. A' is a copy of A.

Iteration for n times:

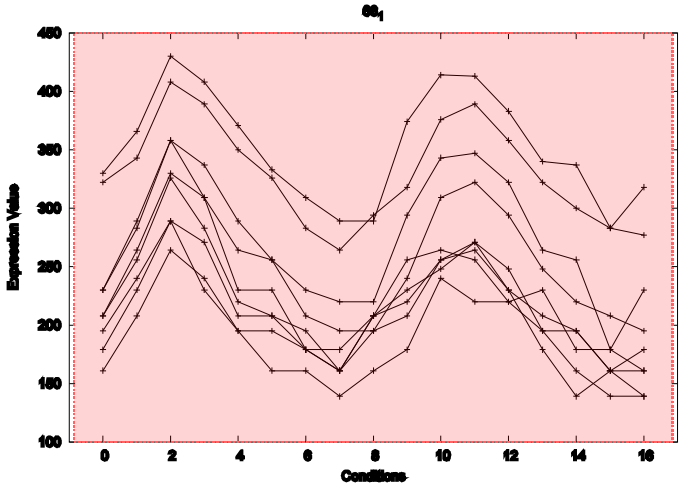
1. Apply Algorithm 2 on A', δ, and α. If the row (column) size is small (less than 100), do not perform multiple node deletion on rows (columns). The matrix after multiple node deletion is B.
2. (Step 5 of Algorithm 2) Apply Algorithm 1 on B and δ and the matrix after single node deletion is C.
3. Apply Algorithm 3 on A and C and the result is the bicluster D.
4. Report D, and replace the elements in A' that are also in D with random numbers.

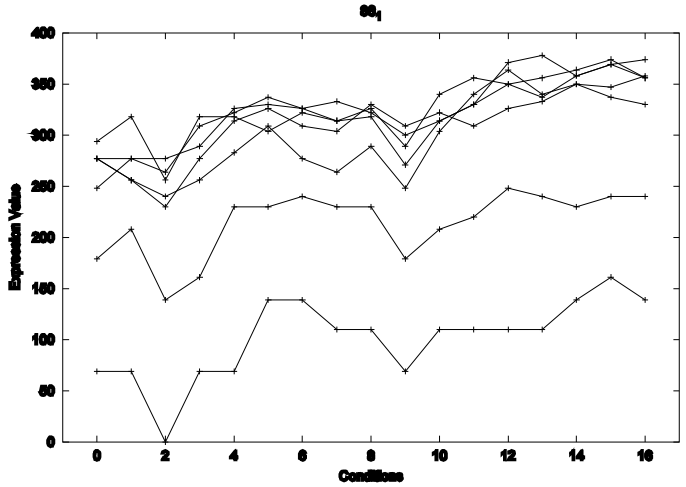


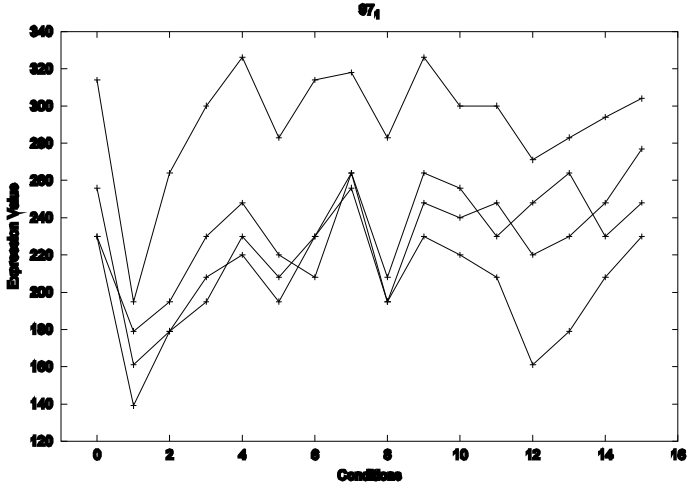
Evolutionary Biclustering

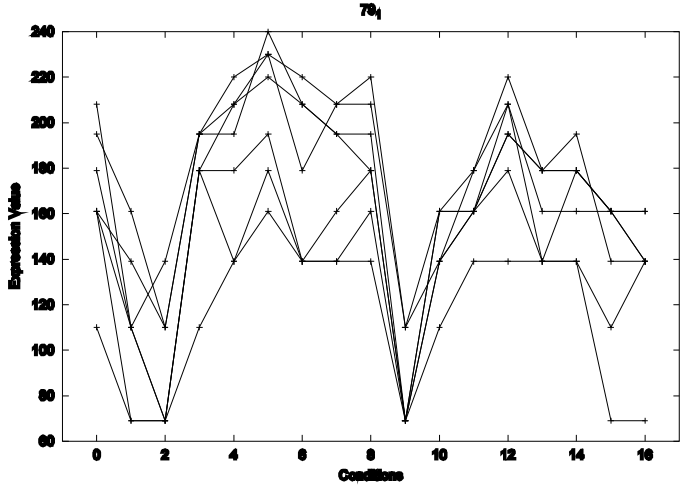
- Binary encoding for rows/columns
- Fitness:
  - mean squared residue
  - row variance
  - large volume
  - penalty (exponential)
- Typical genetic operators

Example: Yeast

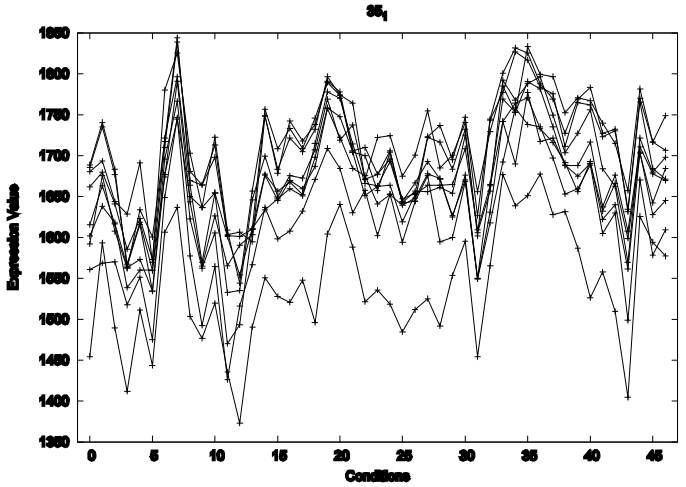


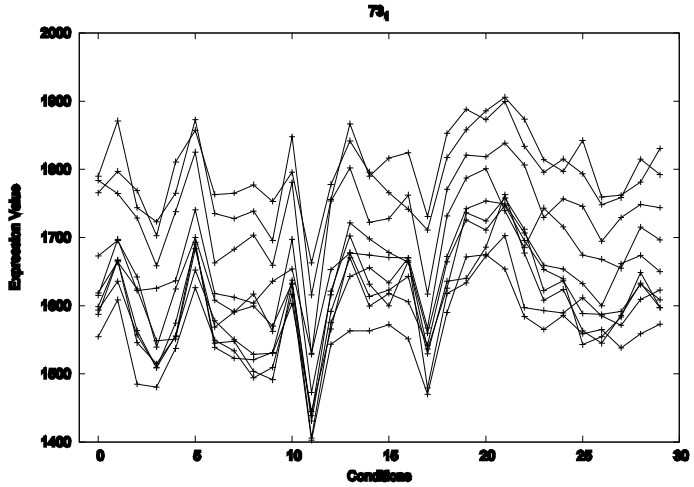


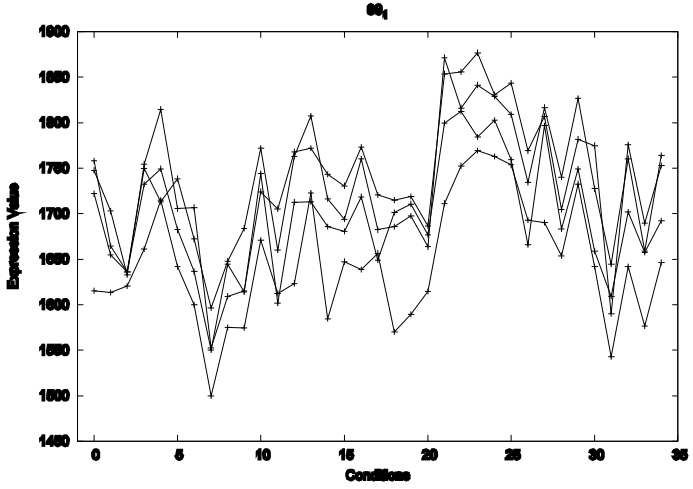


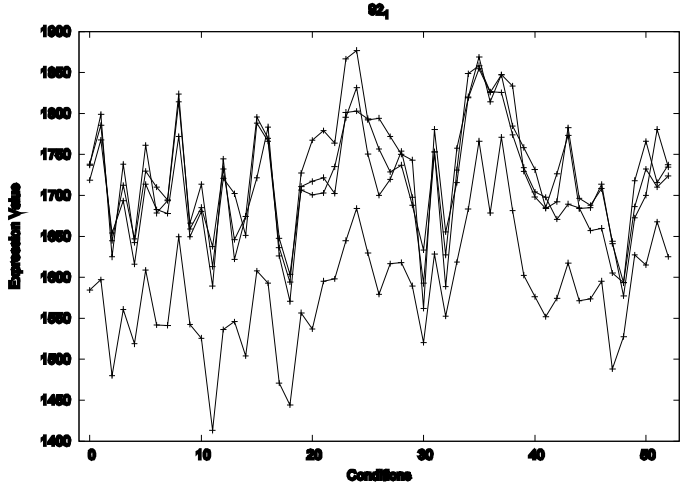


Example: Colon Cancer









Discussion

- Overlapping?
- Sequential covering?
- Local vs. Global strategy?
- Penalty?
- What is the quality measure of a bicluster?
- When a bicluster is better than another one?
- How a bicluster can be statistically validated?
- What knowledge can a bicluster provide?

Applications

- Microarrays
- Collaborative filtering: identify subgroups of customers with similar preferences towards a subset of products
- Recommendation systems for E-commerce
- Information Retrieval: identify subgroups of documents with similar properties relatively to subgroups of attributes, such as words or images (relevant in query and indexing)
- Medline (1033 medical abstracts), Cranfield (1400 aeronautical abstracts), Cisi (1460 information retrieval)
- Electoral data
- Nutritional data

References

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulus, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM/SIGMOD International Conference on Management of Data*, pages 94–105, 1998.

[2] Amir Ben-Dor, Benny Chor, Richard Karp, and Zohar Yakhini. Discovering local structure in gene expression data: The order–preserving submatrix problem. In *Proceedings of the 6th International Conference on Computacional Biology (RECOMB'02)*, pages 49–57, 2002.

[3] Pavel Berkhin and Jonathan Becher. Learning simple relations: theory and applications. In *Proceedings of the 2nd SIAM International Conference on Data Mining*, pages 420–436, 2002.

[4] Stanislav Busygin, Gerrit Jacobsen, and Ewald Kramer. Double conjugated clustering applied o leukemia microarray data. In *Proceedings of the 2nd SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data*, 2002.

[5] Andrea Califano, Gustavo Stolovitzky, and Yunai Tu. Analysis of gene expression microarays for phenotype classification. In *Proceedings of the International Conference on Computacional Molecular Biology*, pages 75–85, 2000.

[6] Yizong Cheng and George M. Church. Biclustering of expression data. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB'00)*, pages 93–103, 2000.

[7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliffoord Stein. *Introduction to Algorithms*. The MIT Electrical Engineering and Computer Science Series. The MIT Press, 2nd edition, 2001.

[8] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 269–274, 2001.

[9] Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretical co-clustering. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, pages 89–98, 2003.

[10] D. Duffy and A. Quiroz. A permutation based algorithm for block clustering. *Journal of Classification*, 8:65–91, 1991.

[11] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. In *Proceedings of the Natural Academy of Sciences USA*, pages 12079–12084, 2000.

[12] Dan Gusfield. *Algorithms on strings, trees, and sequences*. Computer Science and Computational Biology Series. Cambridge University Press, 1997.

[13] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association (JASA)*, 67(337):123–129, 1972.

[14] Jochen Hipp, Ulrich G¨untzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule mining – a general survey and comparison. *SIGKDD Explorations*, 2(1):58–64, July 2000.

[15] Thomas Hofmann and Jaz Puzicha. Latent class models for collaborative filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 668–693, 1999.

References

[16] Yuval Klugar, Ronen Basri, Joseph T. Chang, and Mark Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. In *Genome Research*, volume 13, pages 703–716, 2003.

IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS 30

[17] Laura Lazzeroni and Art Owen. Plaid models for gene expression data. Technical report, Stanford University, 2000.

[18] Jinze Liu and Wei Wang. Op-cluster: Clustering by tendency in high dimensional space. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 187–194, 2003.

[19] T. M. Murali and Simon Kasif. Extracting conserved gene expression motifs from gene expression data. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 8, pages 77–88, 2003.

[20] Ren Peeters. The maximum edge biclique problem is np-complete. *Discrete Applied Mathematics*, 131(3):651–654, 2003.

[21] Eran Segal, Ben Taskar, Audrey Gasch, Nir Friedman, and Daphne Koller. Rich probabilistic models for gene expression. In *Bioinformatics*, volume 17 (Suppl. 1), pages S243–S252, 2001.

[22] Eran Segal, Ben Taskar, Audrey Gasch, Nir Friedman, and Daphne Koller. Decomposing gene expression into cellular processes. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 8, pages 89–100, 2003.

[23] Qizheng Sheng, Yves Moreau, and Bart De Moor. Biclustering micrarray data by gibbs sampling. In *Bioinformatics*, volume 19 (Suppl. 2), pages ii196–ii205, 2003.

[24] Amos Tanay, Roded Sharan, and Ron Shamir. Discovering statistically significant biclusters in gene expression data. In *Bioinformatics*, volume 18 (Suppl. 1), pages S136–S144, 2002.

[25] Chun Tang, Li Zhang, Idon Zhang, and Murali Ramanathan. Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In *Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, pages 41–48, 2001.

[26] R. Tibshirani, T. Hastie, M. Eisen, D. Ross, D. Botstein, and P. Brown. Clustering methods for the analysis of DNA microarray data. Technical report, Department of Health Research and Policy, Department of Genetics and Department of Biochemestry, Stanford University, 1999.

[27] Lyle Ungar and Dean P. Foster. A formal statistical approach to collaborative filtering. In *Proceedings of the Conference on Automated Learning and Discovery (CONALD'98)*, 1998.

[28] Haixun Wang, Wei Wang, Jiong Yang, and Philip S. Yu. Clustering by pattern similarity in large data sets. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 394–405, 2002.

[29] Jiong Yang, Wei Wang, Haixun Wang, and Philip Yu. Æ-clusters: Capturing subspace correlation in a large data set. In *Proceedings of the 18th IEEE International Conference on Data Engineering*, pages 517–528, 2002.

[30] Jiong Yang, Wei Wang, Haixun Wang, and Philip Yu. Enhanced biclustering on expression data. In *Proceedings of the 3rd IEEE Conference on Bioinformatics and Bioengineering*, pages 321–327, 2003.

BICLUSTERING OF GENE EXPRESSION DATA

Thank you

Jesús S. Aguilar-Ruiz
Computer Science Department
University of Seville, SPAIN

http://www.lsi.us.es/~aguilar
aguilar@lsi.us.es