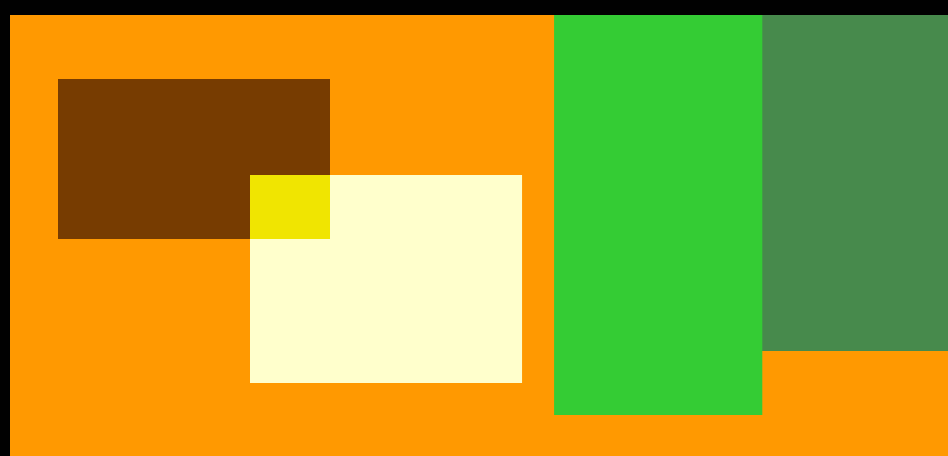
Biclustering Algorithms for Biological Data Analysis

Sara C. Madeira
and Arlindo L. Oliveira

Presentation by
Matthew Hibbs

What is Biclustering?

- Given an n x m matrix, A, find a set of submatrices, B_k, such that the contents of each B_k follow a desired pattern
- Row/Column order need not be consistent between different B_ks

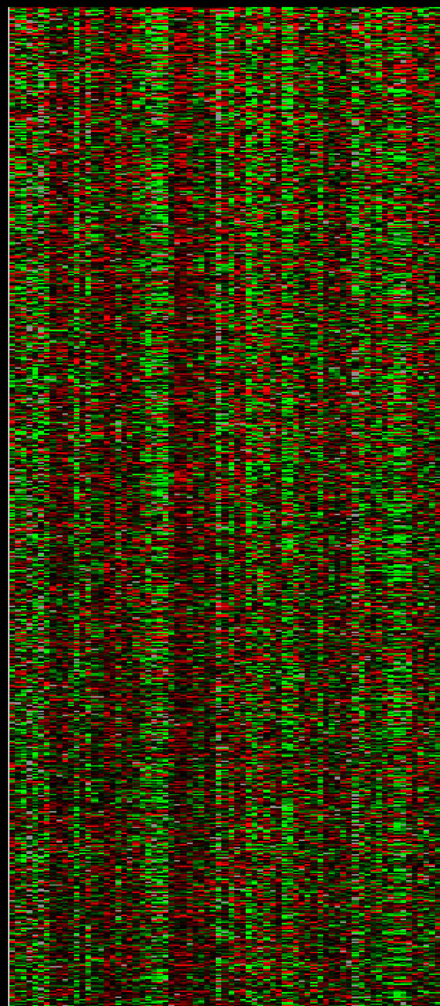


Why Bicluster?

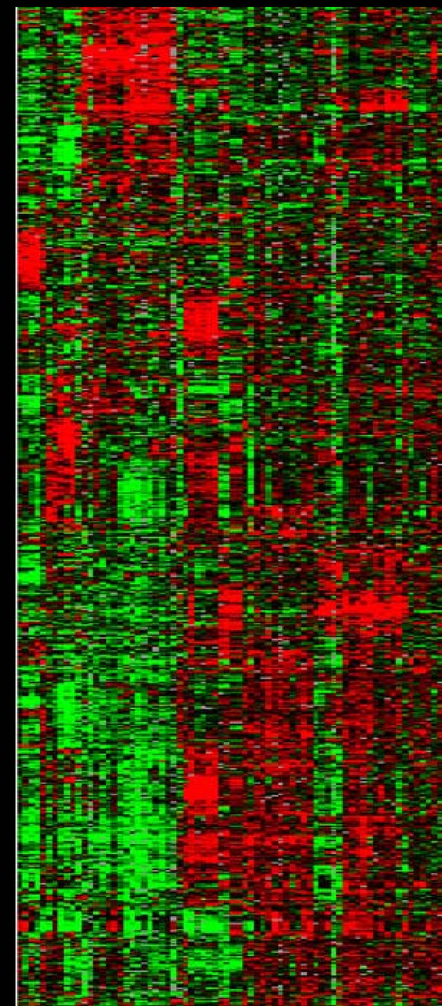
- Genes not regulated under all conditions
- Genes regulated by multiple factors/processes concurrently

- Key to determine function of genes
- Key to determine classification of conditions

Simple "Bicluster"



Unclustered



Clustered

Dataset from Garber et al.

Formal Definitions

- A contains rows X and columns Y
- $X = \{x_1, \ldots, x_n\}$ $Y = \{y_1, \ldots, y_n\}$
- $I \subseteq X$, $J \subseteq Y$, $A_{IJ} = (I,J)$ = submatrix
- $I = \{i_1, \ldots, i_k\}$ $J = \{j_1, \ldots, j_s\}$

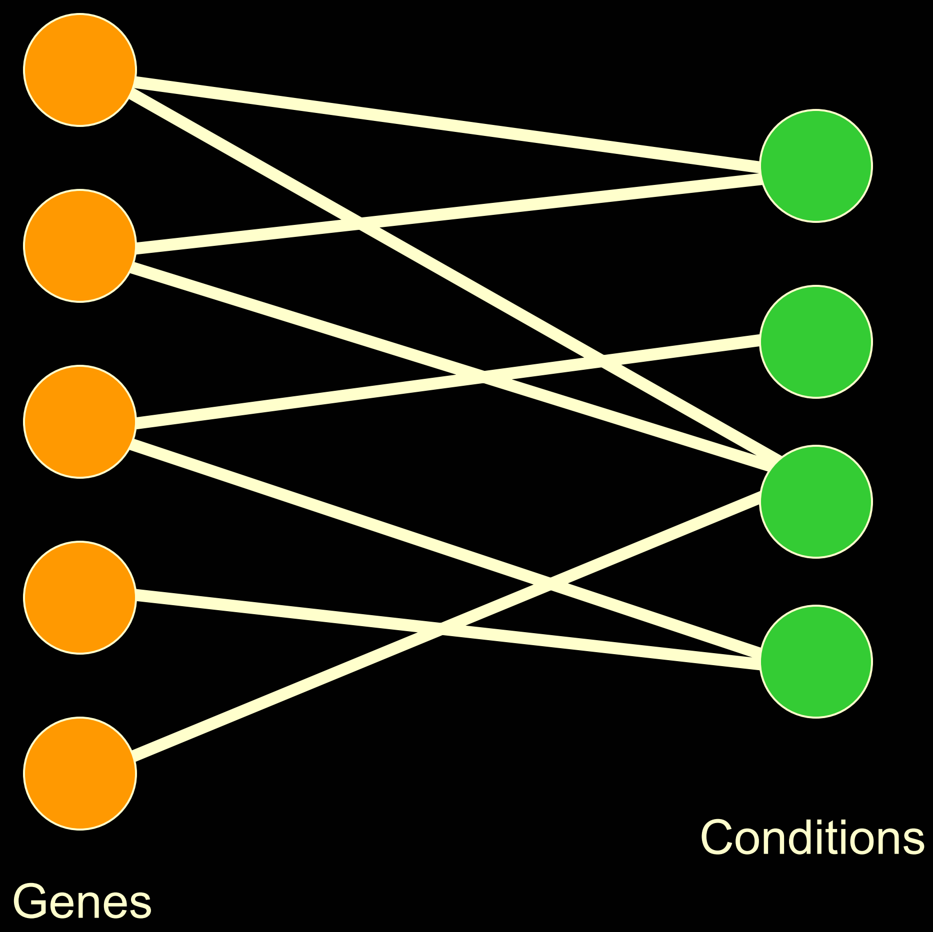
$$a_{iJ} = \frac{1}{|J|}\sum_{j\in J} a_{ij} \qquad a_{IJ} = \frac{1}{|I||J|}\sum_{i\in I, j\in J} a_{ij}$$

$$a_{Ij} = \frac{1}{|I|}\sum_{i\in I} a_{ij} \qquad a_{IJ} = \frac{1}{|I|}\sum_{i\in I} a_{iJ} = \frac{1}{|J|}\sum_{j\in J} a_{Ij}$$

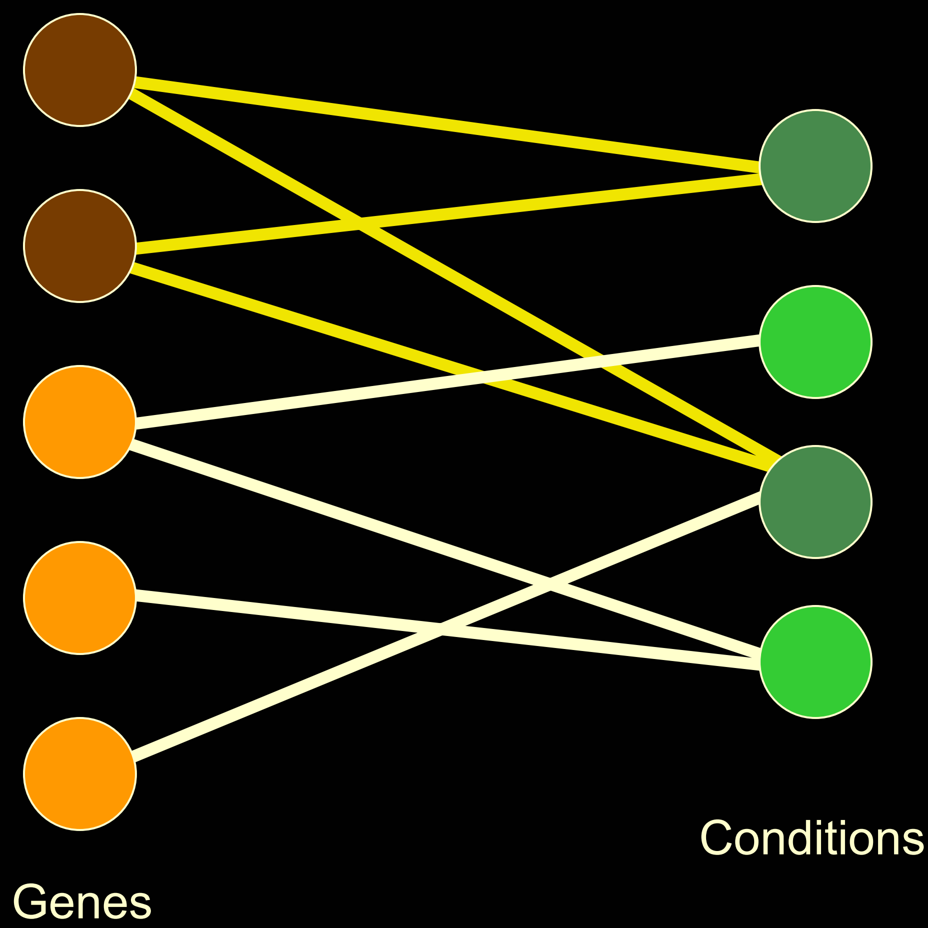
Bipartite Graph

- Matrix can be thought of as a Graph
- Rows are one set of vertices L, Columns are another set R
- Edges are weighted by the corresponding entries in the matrix
- If all weights are binary, biclustering becomes biclique finding

NP-complete



NP-complete



Ordering an Algorithm

- Bicluster Type
- Bicluster Structure
- Algorithmic Approach

Ordering an Algorithm

- Bicluster Type
- Bicluster Structure
- Algorithmic Approach

Bicluster Types

- Constant values
- Constant values on rows or columns
- Coherent values
- Coherent evolutions

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

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

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

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

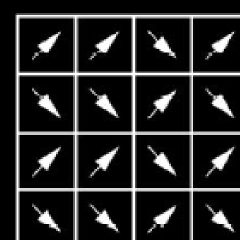
1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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



Constant Values

Perfect: $a_{ij} = \mu$ for all $i \in I$ and $j \in J$

Block Clustering, minimize *variance*:

$$VAR(I,J) = \sum_{i \in I, j \in J} (a_{ij} - a_{IJ})^2.$$

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

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

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

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

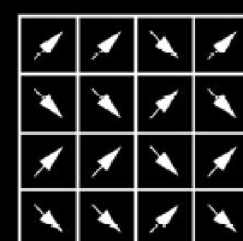
1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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



Constant Values on Rows/Cols

Perfect constant Rows:

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

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

Perfect constant Columns:

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

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

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

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

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

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

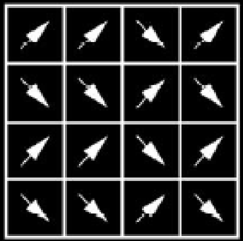
1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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



Constant Values on Rows/Cols

- Normalize rows/cols → constant value
- Noise makes perfect biclusters rare
  - δ-valid ks-patterns (Califano et al.)

$$\max\left(a_{ij}\right) - \min\left(a_{ij}\right) < \delta, \forall j \in J$$

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

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

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

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

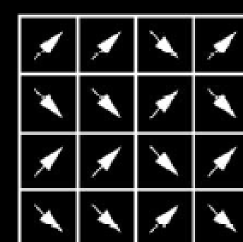
1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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



Coherent values

Perfect additive model:

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

Perfect multiplicative model:

$$a_{ij} = \mu' \times \alpha'_i \times \beta'_j$$

$$\mu = \log(\mu'),\ \alpha_i = \log(\alpha'_i),\ \beta_j = \log(\beta'_j)$$

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

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

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

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

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

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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

↗	↗	↘	↗
↘	↘	↗	↘
↗	↗	↘	↗
↘	↘	↗	↘

Coherent values

Perfect δ-bicluster: $$a_{ij} = a_{iJ} + a_{Ij} - a_{IJ}$$

(Cheng & Church) Residue: $$r(a_{ij}) = a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}$$

Mean squared residue: $$H(I,J) = \frac{1}{|I||J|}\sum_{i\in I, j\in J} r(a_{ij})^2$$

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

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

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

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

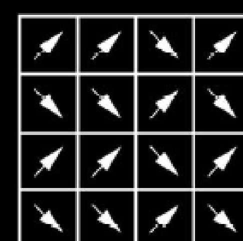
1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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



Overlapping Coherent Values

Plaid Model or General Additive Model (Lazzeroni and Owen)

$$a_{ij} = \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk} \qquad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(a_{ij} - \theta_{ij0} - \sum_{k=1}^{K}\theta_{ijk}\rho_{ik}\kappa_{jk}\right)^2$$

$$\theta_{ijk} = \mu_k,\ \mu_k + \alpha_{ik},\ \mu_k + \beta_{jk},\ \text{or}\ \mu_k + \alpha_{ik} + \beta_{jk}$$

1.0	1.0	1.0	1.0		
1.0	1.0	1.0	1.0		
1.0	1.0	3.0	3.0	2.0	2.0
1.0	1.0	3.0	3.0	2.0	2.0
		2.0	2.0	2.0	2.0
		2.0	2.0	2.0	2.0

1.0	1.0	1.0	1.0		
2.0	2.0	2.0	2.0		
3.0	3.0	8.0	8.0	5.0	5.0
4.0	4.0	10.0	10.0	6.0	6.0
		7.0	7.0	7.0	7.0
		8.0	8.0	8.0	8.0

1.0	2.0	3.0	4.0		
1.0	2.0	3.0	4.0		
1.0	2.0	8.0	10.0	7.0	8.0
1.0	2.0	8.0	10.0	7.0	8.0
		5.0	6.0	7.0	8.0
		5.0	6.0	7.0	8.0

1.0	2.0	5.0	0.0		
2.0	3.0	6.0	1.0		
4.0	5.0	9.0	5.0	5.0	0.0
5.0	6.0	11.0	7.0	6.0	1.0
		4.0	5.0	8.0	3.0
		5.0	6.0	9.0	4.0

Overlapping Coherent Values

Gaussian Distribution with variance depending on row, column, and bicluster variance

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\left(a_{ij}-\theta_{ij0}-\sum_{k=1}^{K}\theta_{ijk}\rho_{ik}\kappa_{jk}\right)^2}{2(\sigma_{iJ}^2+\sigma_{Ij}^2+\sigma_{IJ}^2)}$$

1.0	1.0	1.0	1.0		
1.0	1.0	1.0	1.0		
1.0	1.0	3.0	3.0	2.0	2.0
1.0	1.0	3.0	3.0	2.0	2.0
		2.0	2.0	2.0	2.0
		2.0	2.0	2.0	2.0

1.0	1.0	1.0	1.0		
2.0	2.0	2.0	2.0		
3.0	3.0	8.0	8.0	5.0	5.0
4.0	4.0	10.0	10.0	6.0	6.0
		7.0	7.0	7.0	7.0
		8.0	8.0	8.0	8.0

1.0	2.0	3.0	4.0		
1.0	2.0	3.0	4.0		
1.0	2.0	8.0	10.0	7.0	8.0
1.0	2.0	8.0	10.0	7.0	8.0
		5.0	6.0	7.0	8.0
		5.0	6.0	7.0	8.0

1.0	2.0	5.0	0.0		
2.0	3.0	6.0	1.0		
4.0	5.0	9.0	5.0	5.0	0.0
5.0	6.0	11.0	7.0	6.0	1.0
		4.0	5.0	8.0	3.0
		5.0	6.0	9.0	4.0

Coherent Evolutions

- Order-Preserving Submatrix (OPSM, Ben-Dor et al.)
- Order-Preserving Cluster (OP-Cluster, Liu and Wang)
- Rank-based approaches

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

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

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

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

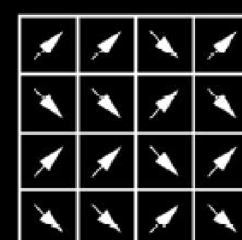
1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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



Coherent Evolutions

- Quantized expression levels into states
- Maximize conserved rows/cols (Murali and Kasif)

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

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

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

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

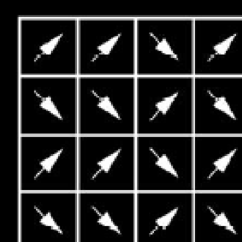
1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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



Coherent Evolutions

- Identify “significant conditions” with respect to normal levels
- Construct a bipartite graph containing only these edges
- bicluster is biclique (simple SAMBA, Tanay et al.)

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

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

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

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

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

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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

↗	↗	↘	↗
↘	↘	↗	↘
↗	↗	↘	↗
↘	↘	↗	↘

Coherent Evolutions

- Label "significant conditions" as up or down regulated
- Select columns that have the same or opposite effect
- (refined SAMBA, Tanay et al.)

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

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

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

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

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

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

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

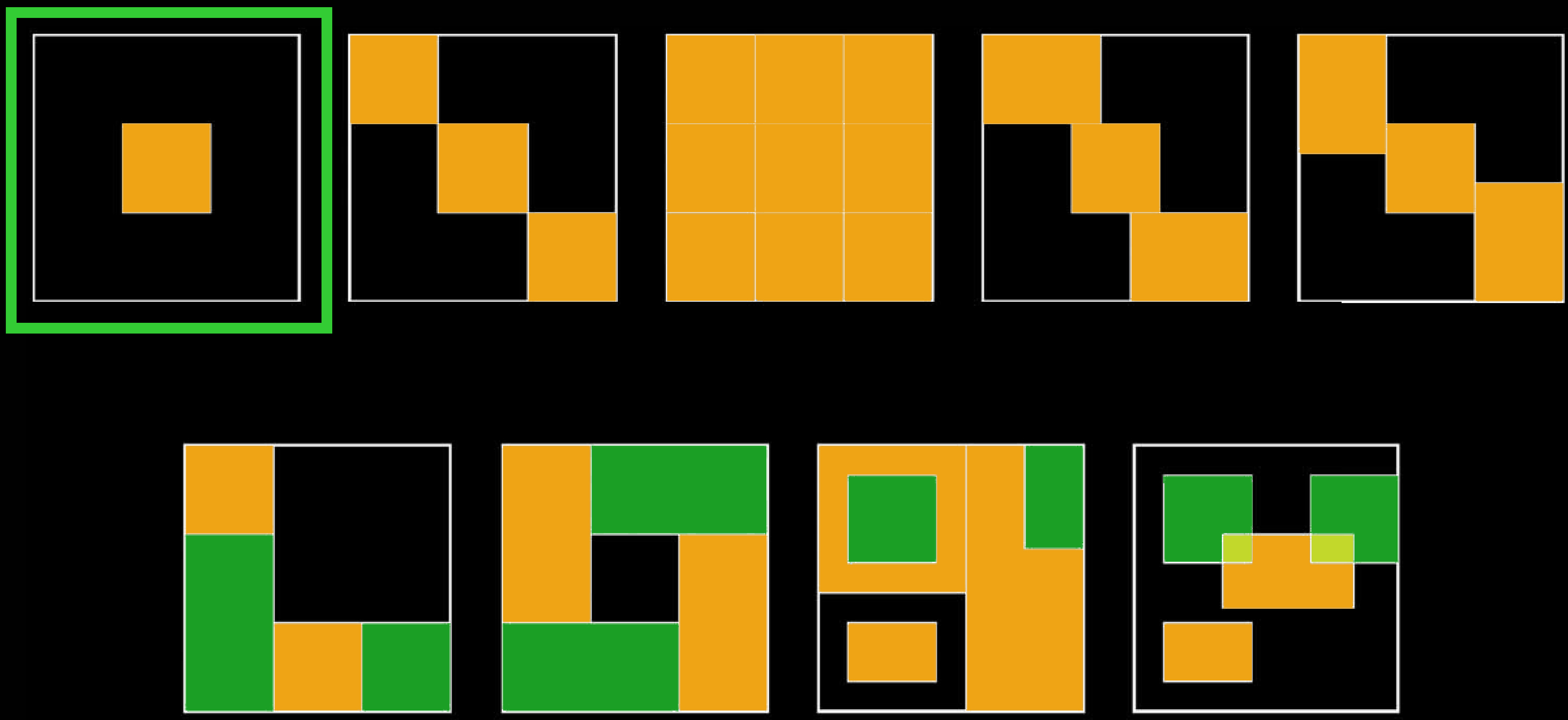
↗	↗	↘	↗
↘	↘	↗	↘
↗	↗	↘	↗
↘	↘	↗	↘

Ordering an Algorithm

- Bicluster Type
- Bicluster Structure
- Algorithmic Approach

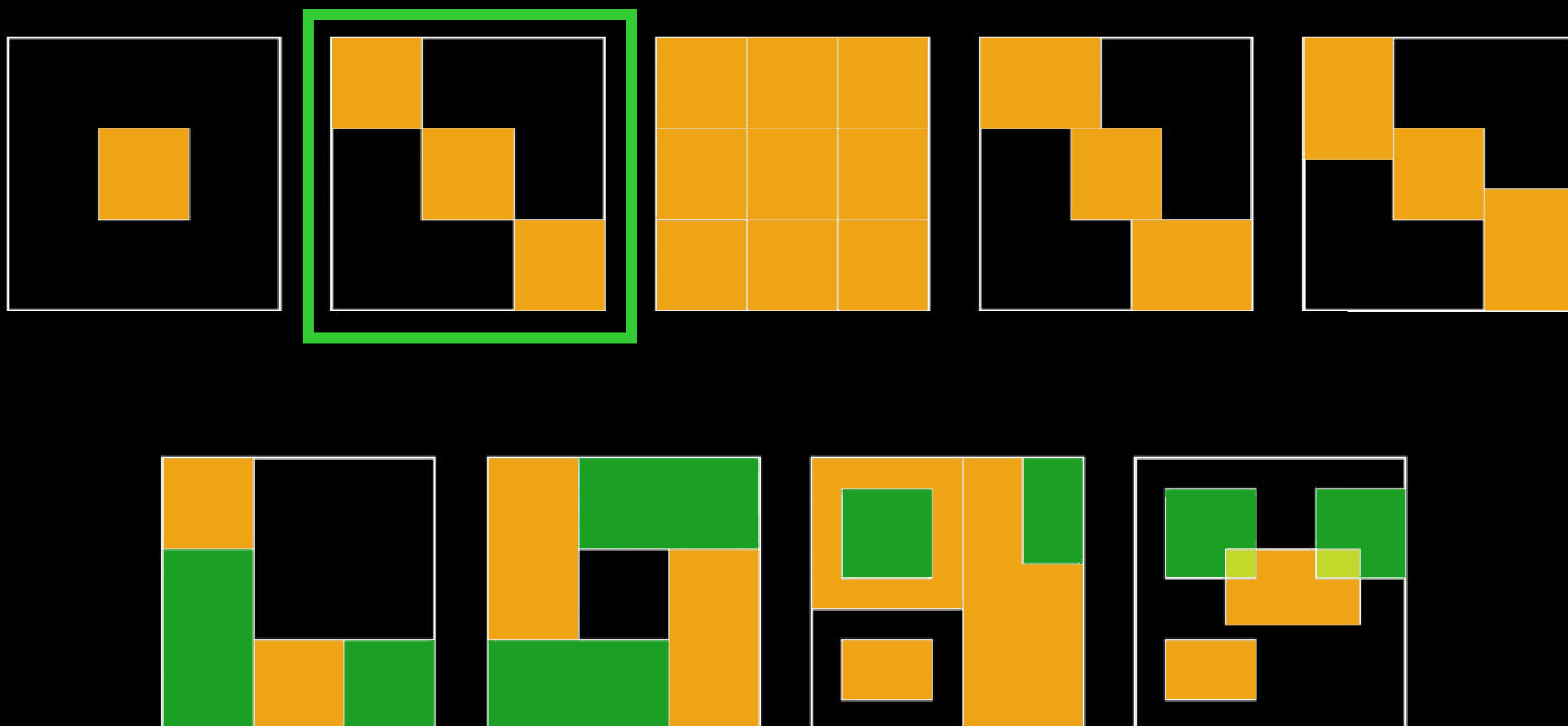
Bicluster Structure

Single Bicluster



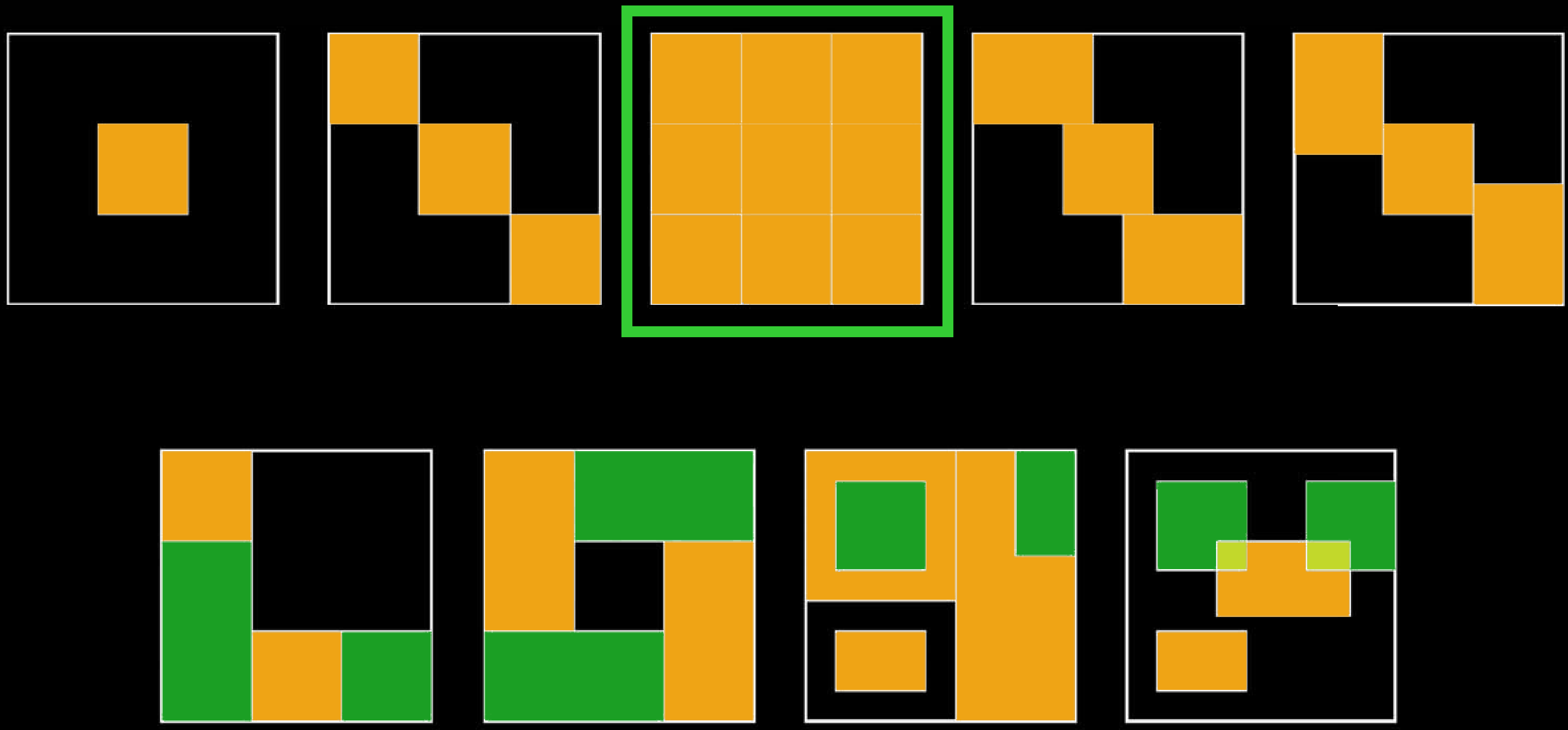
Bicluster Structure

Exclusive row and column bicluster



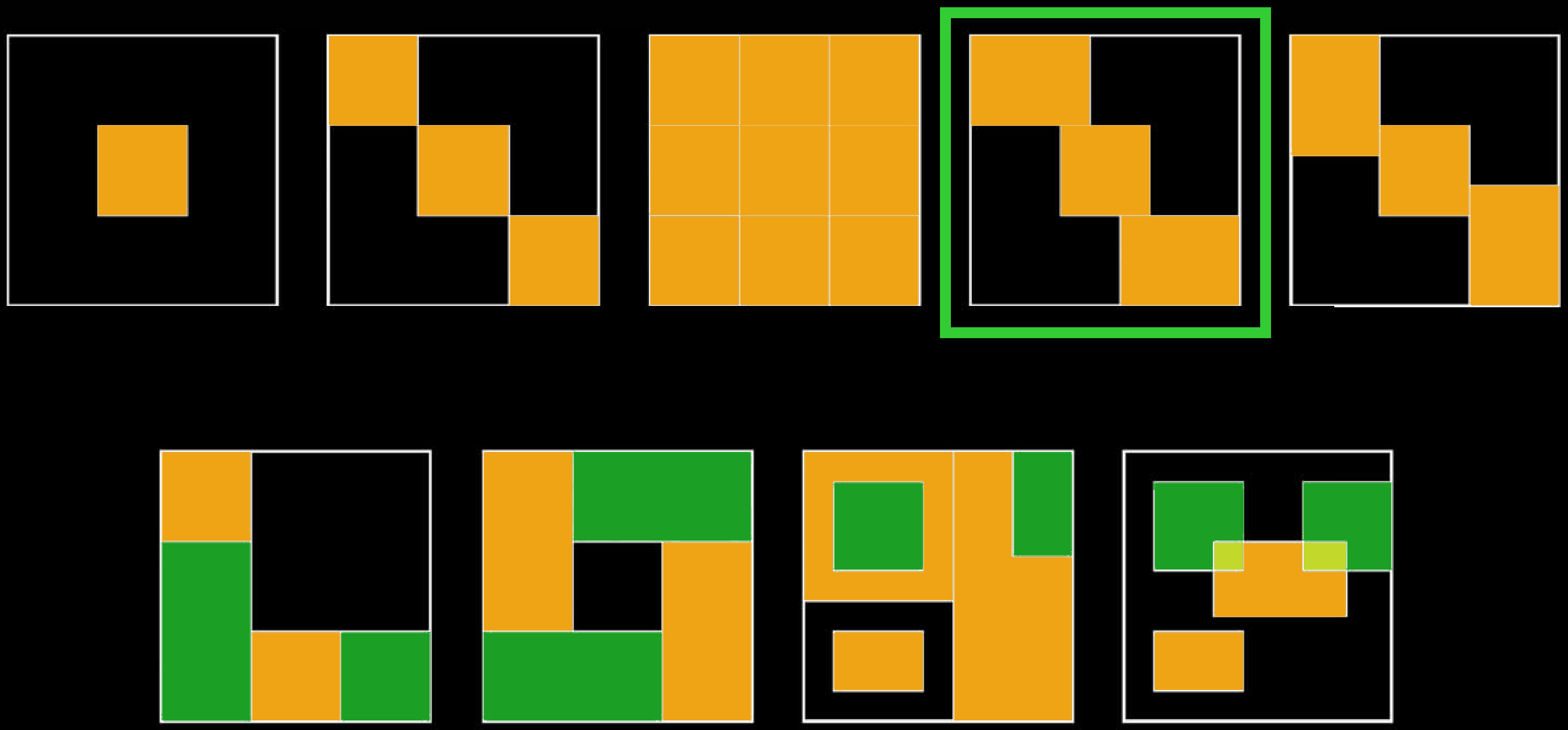
Bicluster Structure

Non-overlapping checkerboard biclusters



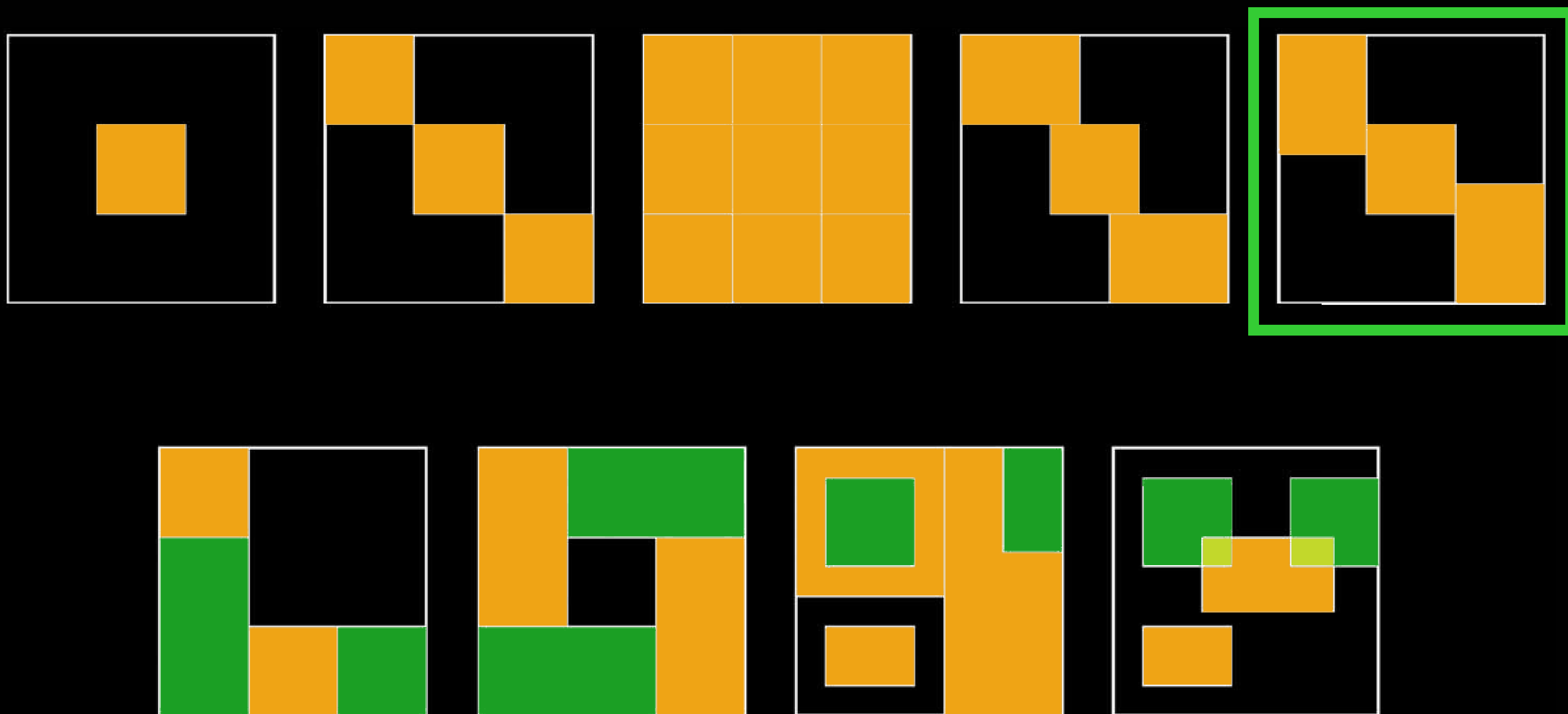
Bicluster Structure

Exclusive-rows biclusters



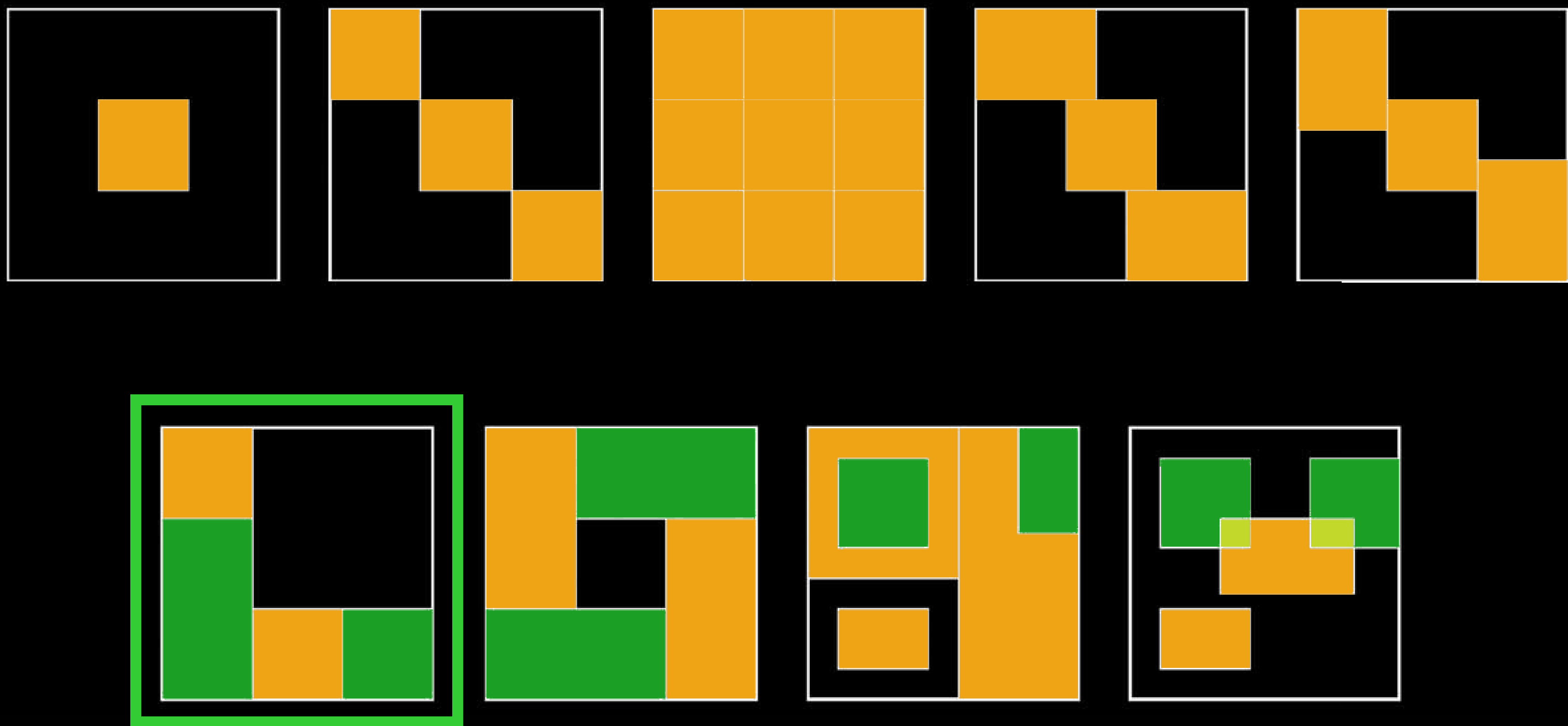
Bicluster Structure

Exclusive-columns biclusters



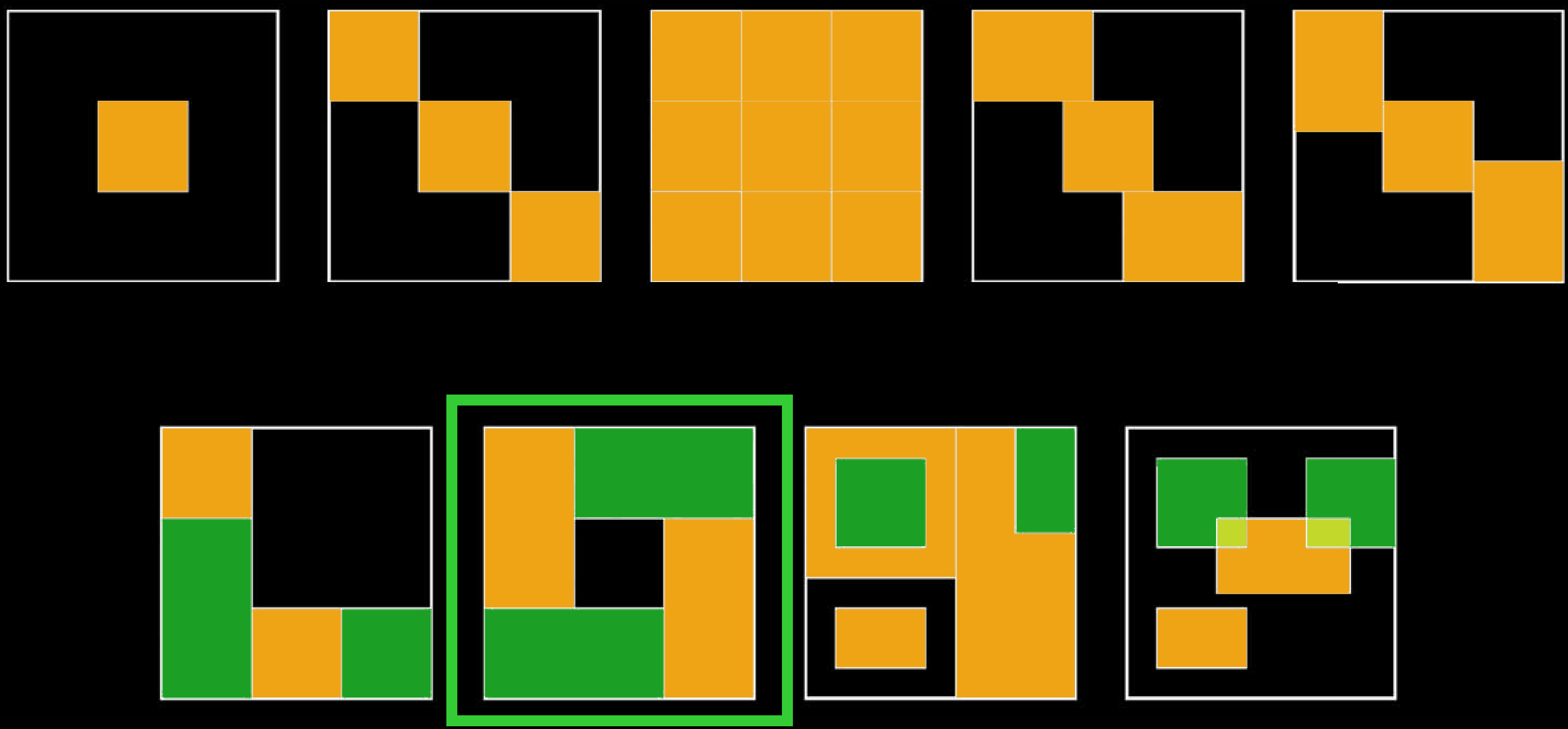
Bicluster Structure

Non-overlapping tree structured biclusters



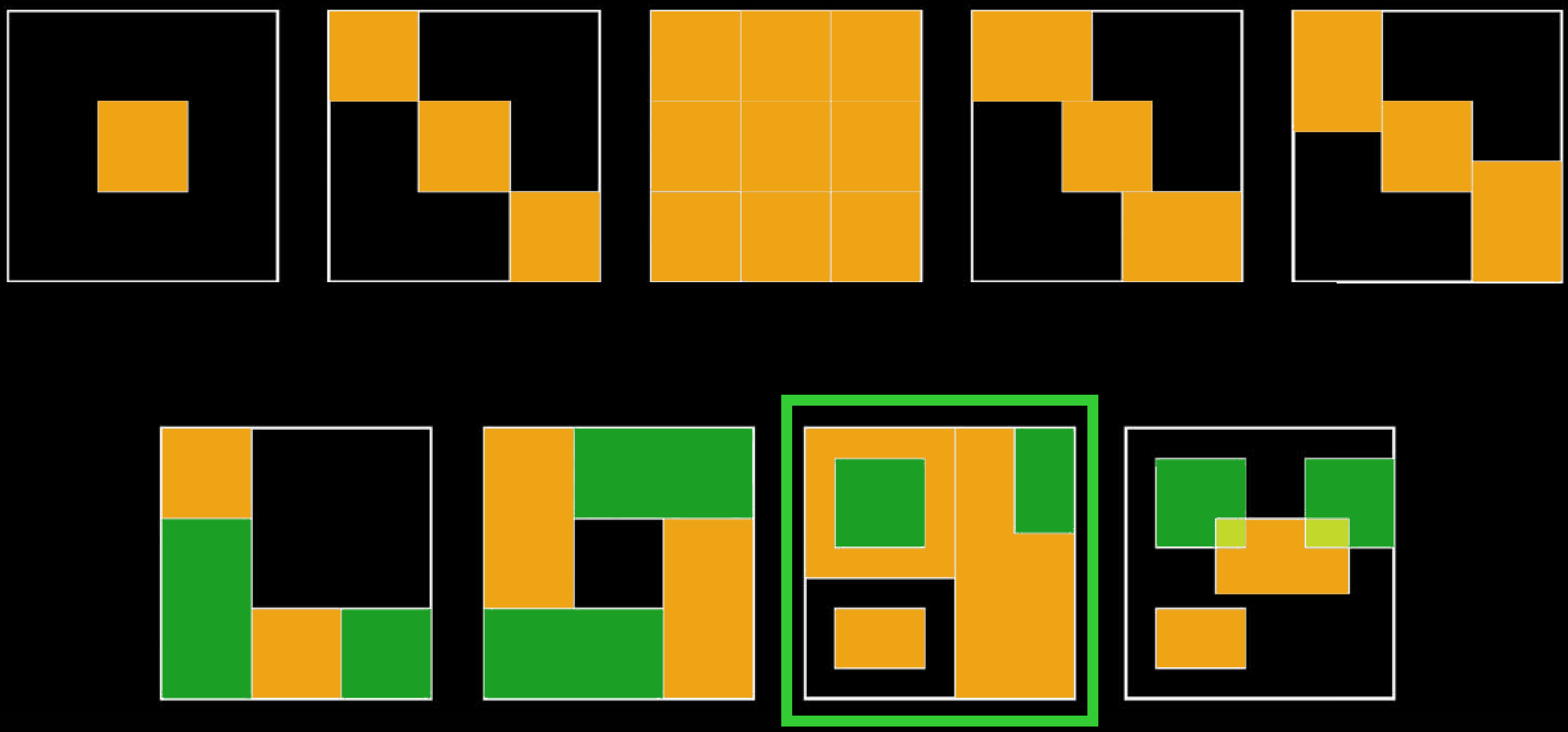
Bicluster Structure

Non-overlapping non-exclusive biclusters



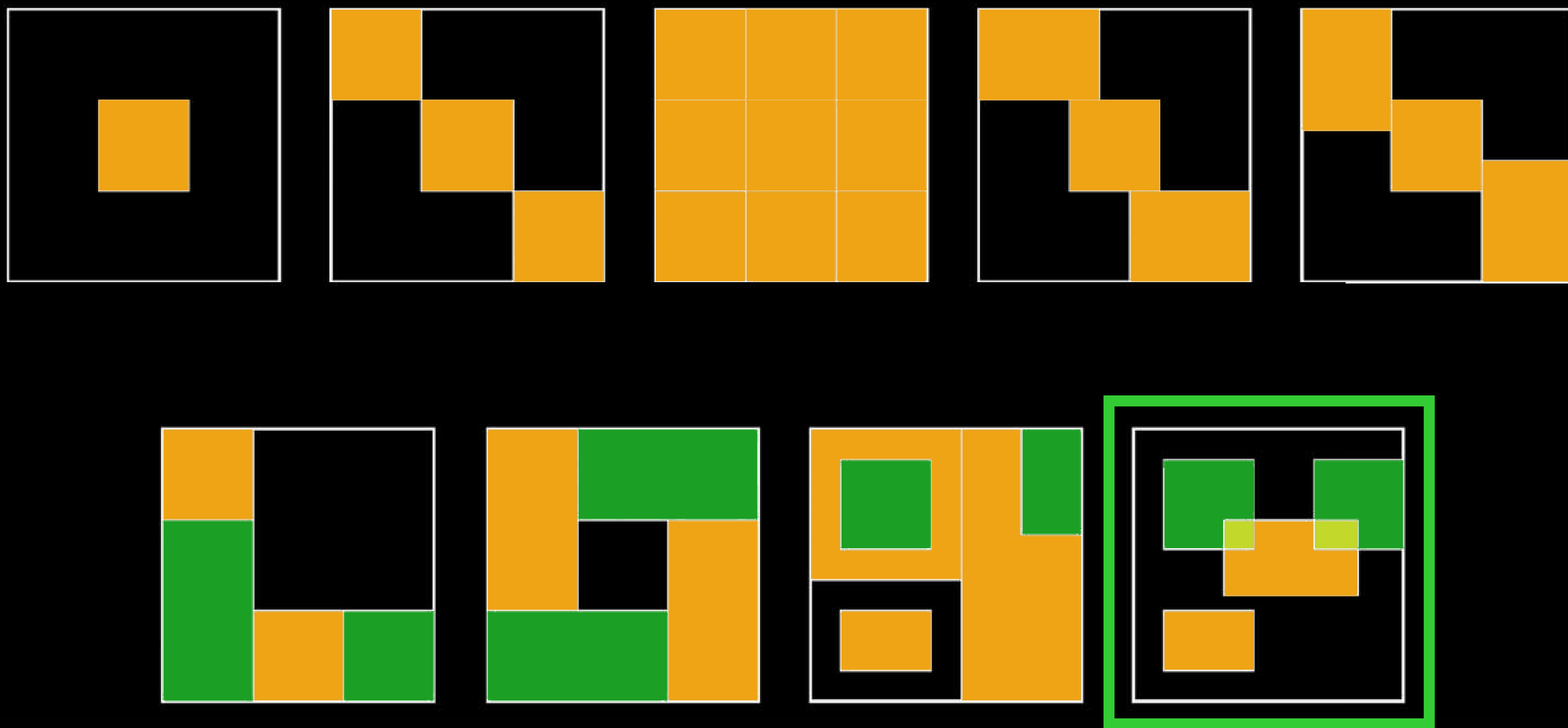
Bicluster Structure

Overlapping hierarchically-structured biclusters



Bicluster Structure

Arbitrary overlapping biclusters



Ordering an Algorithm

- Bicluster Type
- Bicluster Structure
- Algorithmic Approach

Algorithmic Approaches

- Iterative row and column clustering combo
- Divide and conquer
- Greedy iterative search
- Exhaustive bicluster enumeration
- Distribution parameter identification

Iterative Row and Column

- Coupled Two-Way Clustering (CTWC)
  - Keep sets of row and column clusters
  - Start with all rows and all columns
  - Hierarchically cluster based on pairs of row/column clusters
  - Identify new “stable” row/column clusters
  - Repeat until a desired resolution

Iterative Row and Column

- Interrelated Two-Way Clustering (ITWC)
  - Cluster the rows into K groups
  - Cluster columns into two groups based on each row group
  - Combine row and column clusterings
  - Identify row/column cluster pairs that are very different from each other
  - Keep the best 1/3 rows in the heterogeneous pairs
  - Repeat

Divide and Conquer

- Block Clustering (Hartigan)
  - Sort by row or column mean
  - Find best row or column split to reduce “within block” variance
  - Continue, alternating row or column splits
  - Stop when arrive at desired K blocks
- Very fast, but likely to miss good biclusters due to early splits

Greedy Iterative Search

- δ-biclusters (Cheung & Church)
  - Find biclusters with mean squared residue < δ
  - Iterative procedure
    - Remove the row/col that reduces H the most
    - Add rows/cols that do not increase H
  - Stop when H < δ
  - Mask bicluster with random values
  - Repeat to find next bicluster

Exhaustive Bicluster Enumeration

- Statistical-Algorithmic Method for Bicluster Analysis (SAMBA, Tanay et al.)
  - Conversion to bipartite graph
  - Equivalent to selection of heaviest subgraphs
    - Assumes rows have d-bounded degree
  - Report the K heaviest bicliques

Distribution Parameter Identification

- Plaid Model (Lazzeroni and Owen)
  - Given K-1 biclusters, select the Kth bicluster that minimizes sum of squared errors

$$Q = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}(Z_{ij} - \theta_{ijK}\rho_{iK}\kappa_{jK})^2$$

$$Z_{ij} = a_{ij} - \theta_{ij0} - \sum_{k=1}^{K-1}\theta_{ijk}\rho_{ik}\kappa_{jk}$$

  - Sort of like EM, determine θ from ρ and κ, then ρ from θ and κ, then κ from θ and ρ
  - "Fuzzy" membership until end (ρ, κ $\in$ [0,1])

Ordering an Algorithm

- Bicluster Type
- Bicluster Structure
- Algorithmic Approach

Comparison of Biclustering Algorithms

		Type	Structure	Discovery	Approach
Block Clustering	[24]	Constant	4(f)	One Set at a Time	Div-and-Conq
δ-biclusters	[10]	Coherent Values	4(i)	One at a Time	Greedy
FLOC	[50]	Coherent Values	4(i)	Simultaneous	Greedy
FLOC	[51]	Coherent Values	4(i)	Simultaneous	Greedy
pClusters	[48]	Coherent Values	4(g)	Simultaneous	Exh-Enum
Plaid Models	[34]	Coherent Values	4(i)	One at a Time	Dist-Ident
PRMs	[41]	Constant Columns	4(b)	Simultaneous	Dist-Ident
PRMs	[40]	Coherent Values	4(i)	Simultaneous	Dist-Ident
CTWC	[21]	Coherent Values	4(i)	One Set at a Time	Clust-Comb
ITWC	[45]	Coherent Values	4(d)/4(e)	One Set at a Time	Clust-Comb
DCC	[8]	Coherent Values	4(b)/4(c)	Simultaneous	Clust-Comb
δ-Patterns	[9]	Constant Rows	4(i)	Simultaneous	Greedy
Spectral	[32]	Coherent Values	4(c)	Simultaneous	Greedy
Gibbs	[42]	Constant Columns	4(d)/4(e)	One at a Time	Dist-Ident
OPSMs	[6]	Coherent Evolution	4(a)/4(i)	One at a Time	Greedy
SAMBA	[44]	Coherent Evolution	4(i)	Simultaneous	Exh-Enum
xMOTIFs	[37]	Coherent Evolution	4(a)/4(i)	Simultaneous	Greedy
OP-Clusters	[35]	Coherent Evolution	4(i)	Simultaneous	Exh-Enum
Co-Clusters	[11]	Constant/Coherent Values	4(i)	Simultaneous	Greedy

Other Applications

- Information Retrieval
  - Targeted marketing
  - Text mining
- Dimensionality Reduction
  - automatic subspace clustering
- Electoral Data Analysis
- much more…

Conclusions

- Biclustering is useful for BioInformatics and many other research areas
- NP-Complete
- Choice of bicluster type and structure with an algorithmic approach → Algorithm