VII International Course of Massive Data Analysis, MDA 2011

(Clustering and) Biclustering Gene Expression Data

Sara C. Madeira

http://web.ist.utl.pt/sara.madeira/

INSTITUTO SUPERIOR TÉCNICO

inesc id lisboa kdbio

Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento em Lisboa

Knowledge Discovery and Bioinformatics

Roadmap

- Clustering (just a very small overview!!)
  - Hierarchical
  - Partitional
- Biclustering
- Biclustering Gene Expression Time Series

Clustering

- In machine learning, clustering is an Unsupervised Learning technique (no predefined classes or labeled training examples are used).
- Can be Used:
  - As a stand-alone tool to gain insight into the distribution of data, to observe the characteristics of each cluster.
  - As a preprocessing step for classification algorithms, which would then operate on the detected clusters .
- Widely used in numerous applications:
  - Pattern Recognition, Image Processing
  - Market Research, Customer Segmentation
  - Analysis of gene expression data
  - ...

Clustering

- Suppose the data set to be clustered contains **N** objects.
- Objects may be customers, genes, …
- Most clustering algorithms use one of the following **data structures**:
  - Data Matrix (Object-by-Attribute structure)
  - Dissimilarity Matrix (Object-by-Object structure)
- The Data Matrix is often called a *Two-Mode Matrix* since the rows and the columns represents different entities.
- The Dissimilarity Matrix is often called a *One-Mode Matrix* since the rows and the columns represents the same entity.

Clustering

- Represents **N** objects with **M** attributes (also called variables, features, measurements, …).
- *When clustering Gene Expression Data*
  - The N objects can be genes, and the M attributes can be the conditions: condition 1, condition 2, …., or vice versa.

$$X_{(N,M)} = \begin{array}{cc} \text{ATTRIBUTES} & \\ \begin{bmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{iM} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{N1} & \cdots & x_{Nj} & \cdots & x_{NM} \end{bmatrix} & \text{OBJECTS} \end{array}$$

Clustering

$$\begin{array}{ccc} & \text{OBJECTS} & \\ D_{(N,N)} = & \begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(N,1) & d(N,2) & \dots & \dots & 0 \end{bmatrix} & \text{OBJECTS} \end{array}$$

- $d(i,j)$ is the dissimilarity/difference between objects *i* and *j*.
- In general $d(i,j) \in [0,\infty[$ and is close to 0 when objects *i* and *j* are highly similar or "near" each other, and becomes larger the more they differ.
- Most dissimilarity measures are based on a geometric distance and their computation depends on type of attributes.

Clustering

- **Distances used when clustering expression data are related to**
  - Absolute differences (Euclidean distance, …)
  - Trends (Pearson Correlation, …)

- **Homogeneity and Separation Principles should be preserved!!**
  - **Homogeneity:** Genes/conditions within a cluster are close/correlated to each other
  - **Separation:** Genes/conditions in different clusters are further apart from each other/uncorrelated to each other

→ clustering is not an easy task!

Clustering Techniques

- **Agglomerative**

  Start with every gene/condition in its own cluster, and iteratively join clusters together.

- **Divisive**

  Start with one cluster and iteratively divide it into smaller clusters.

- **Hierarchical**

  Organize elements into a tree, leaves represent genes and the length of the pathes between leaves represents the distances between genes/conditions. Similar genes/conditions lie within the same subtrees.

- **Partitional**

  Partitions the genes/conditions into a specified number of groups.

Hierarchical Clustering

- Groups data objects into a tree of clusters (**Dendogram**).
- Bottom-Up: *Agglomerative* Clustering
  - Starts by placing each object in its own cluster.
  - At each step merges the two most similar clusters.
  - Stops when all the objects are in a single cluster or certain termination criteria is satisfied.
- Top-Down: *Divisive* Clustering
  - Starts by placing all the objects in one cluster.
  - At each step splits a cluster into two new clusters.
  - Stops when all the objects are in its own cluster or a termination criteria is satisfied.

Once a split or a merge is made it is impossible to go back!

Hierarchical Clustering

- Most hierarchical clustering algorithms are *Agglomerative.*

- Main difference is on the definition of intercluster similarity:

  - Single Link: Distance between two clusters is the distance between the two closest pair of objects.

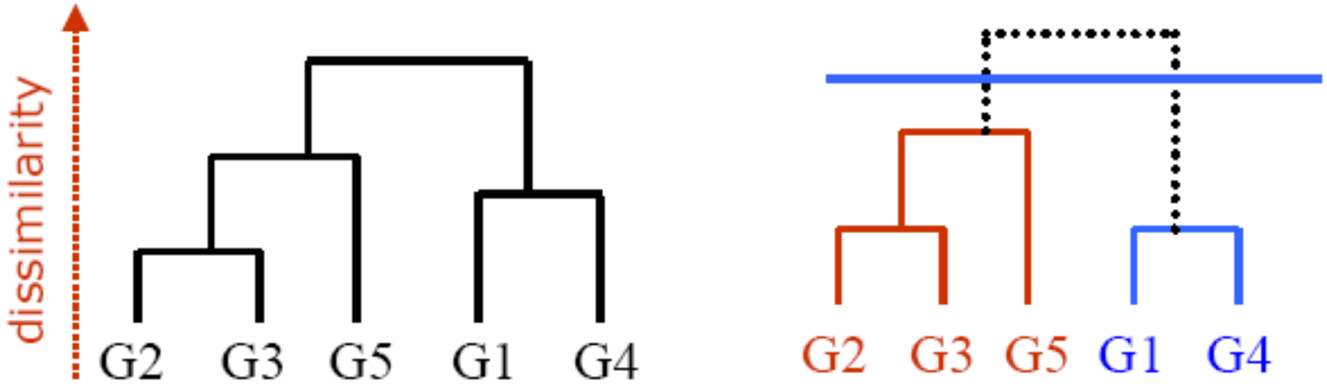
  - Complete Link: Distance between two clusters is the distance between the two farthest pair of objects.

  - Average Link: Distance between two clusters is the average distance between all pairs of object in the two clusters.

Hierarchical Clustering

- Hierarchical clustering does not produce clusters.
- A **Dendogram** is the result of hierarchical clustering.
- Cutting the Dendogram at a certain level yields clusters.
- Each object belongs exactly to one cluster.

Dendogram cutting is a problem analogous to the selection of K in Partitional Clustering algorithms!



Partitional Clustering

- Given a database of **N** objects, partition the objects into a *pre-specified* number of **K** clusters.
- The clusters are formed to optimize a *similarity function*:
  - Intra-cluster similarity must be high.
  - Inter-cluster similarity must be low.
- Each object belongs exactly to one cluster.
- Popular Partitioning Algorithms
  - k-Means
  - EM (Expectation Maximization)

Previous specification of **k** is difficult!

Clustering in Babelomics

- **Algorithms:** UGMA, k-Means, SOTA (Dopazo and Carazo, 1997; Herrero et al., 2001)

- **Webpage:** *http://babelomics.bioinfo.cipf.es/*

- **Tutorial***: http://bioinfo.cipf.es/babelomicstutorial/clustering/*

Roadmap

- Clustering
- **Biclustering**
  - **Why Biclustering and not just Clustering?**
  - Bicluster Types and Structure
  - Algorithms
- Biclustering Gene Expression Time Series

What is Biclustering?

- **Simultaneous Clustering of both rows and columns of a data matrix.**
  - **Biclustering** - Identifies groups of genes with similar/coherent expression patterns under a **specific subset of the conditions**.
  - **Clustering** - Identifies groups of genes/conditions that show similar activity patterns under **all the set of conditions/all the set of genes** under analysis**.**
- **|R| by |C| data matrix** $A = (R,C)$
  - $R=\{r_1,\ldots, r_{|R|}\}$ = Set of |R| rows.
  - $C=\{y_1,\ldots, y_{|C|}\}$ = Set of |C| columns.
  - a_{ij}=relation between row *i* and column *j*.
- **Gene expression matrices**
  - ***R* = Set of Genes**
  - **C = Set of Conditions.**
  - **a_{ij} = expression level of gene *i* under condition *j* (quantity of mRNA).**

	Cond 1	…	Cond *j*	…	Cond.*m*
Gene 1	…	…	…	…	…
…	…	…	…	…	…
Gene *i*	…	…	a_{ij}	…	…
…	…	…	…	…	…
Gene *n*	…	…	…	…	…

Biclustering *vs* Clustering

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
G_1	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{110}
G_2	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}	a_{210}
G_3	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}	a_{39}	a_{310}
G_4	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}	a_{49}	a_{410}
G_5	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}	a_{59}	a_{510}
G_6	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}	a_{69}	a_{610}

$R = \{G_1, G_2, G_3, G_4, G_5, G_6\}$

$C = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}\}$

$I = \{G_2, G_3, G_4\}$

$J = \{C_4, C_5, C_6\}$

Cluster of Conditions (R,J)

$(R, \{C_4, C_5, C_6\})$

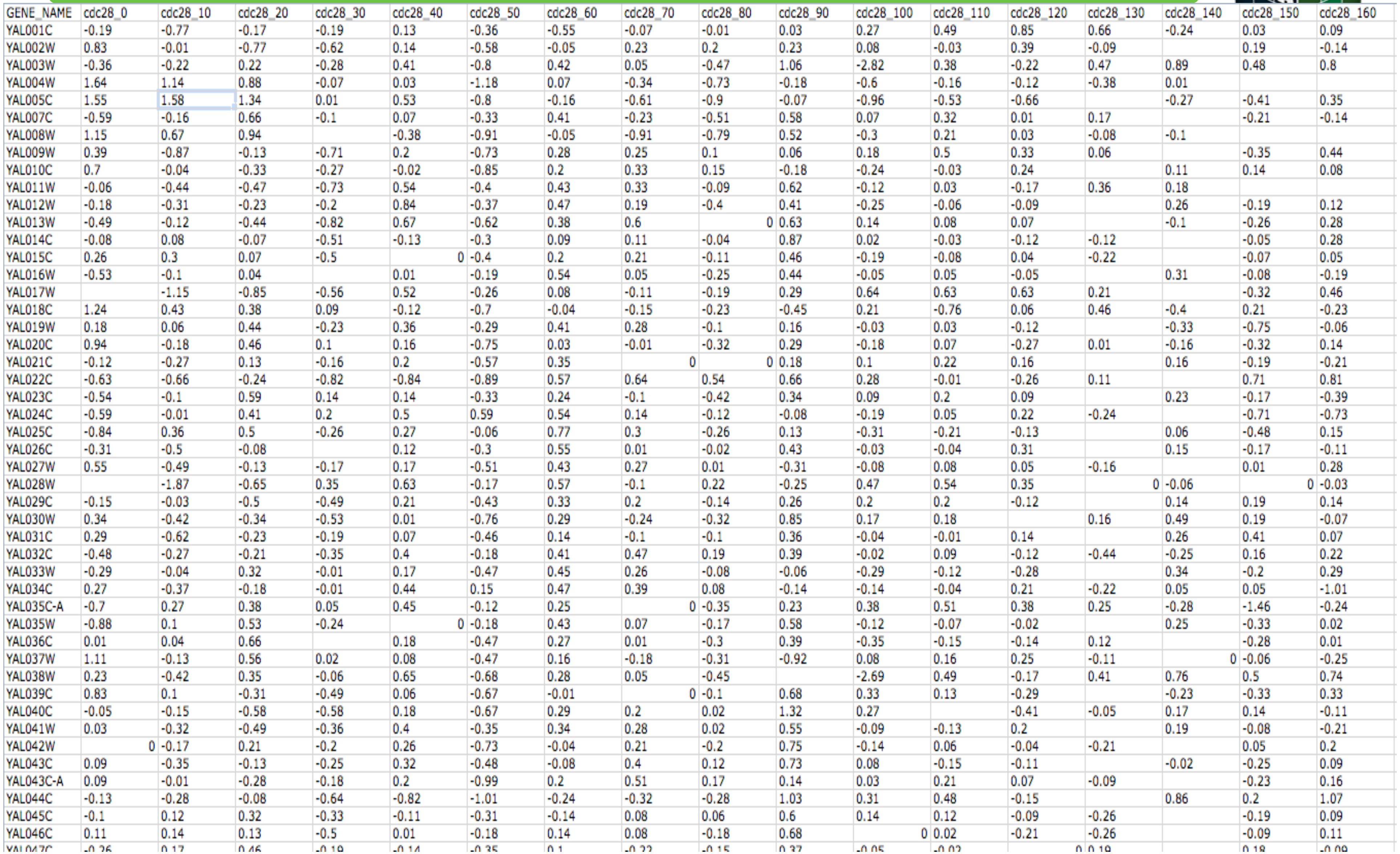
Cluster of Genes (I,C)

$(\{G_2, G_3, G_4\}, C)$

Bicluster (I,J)

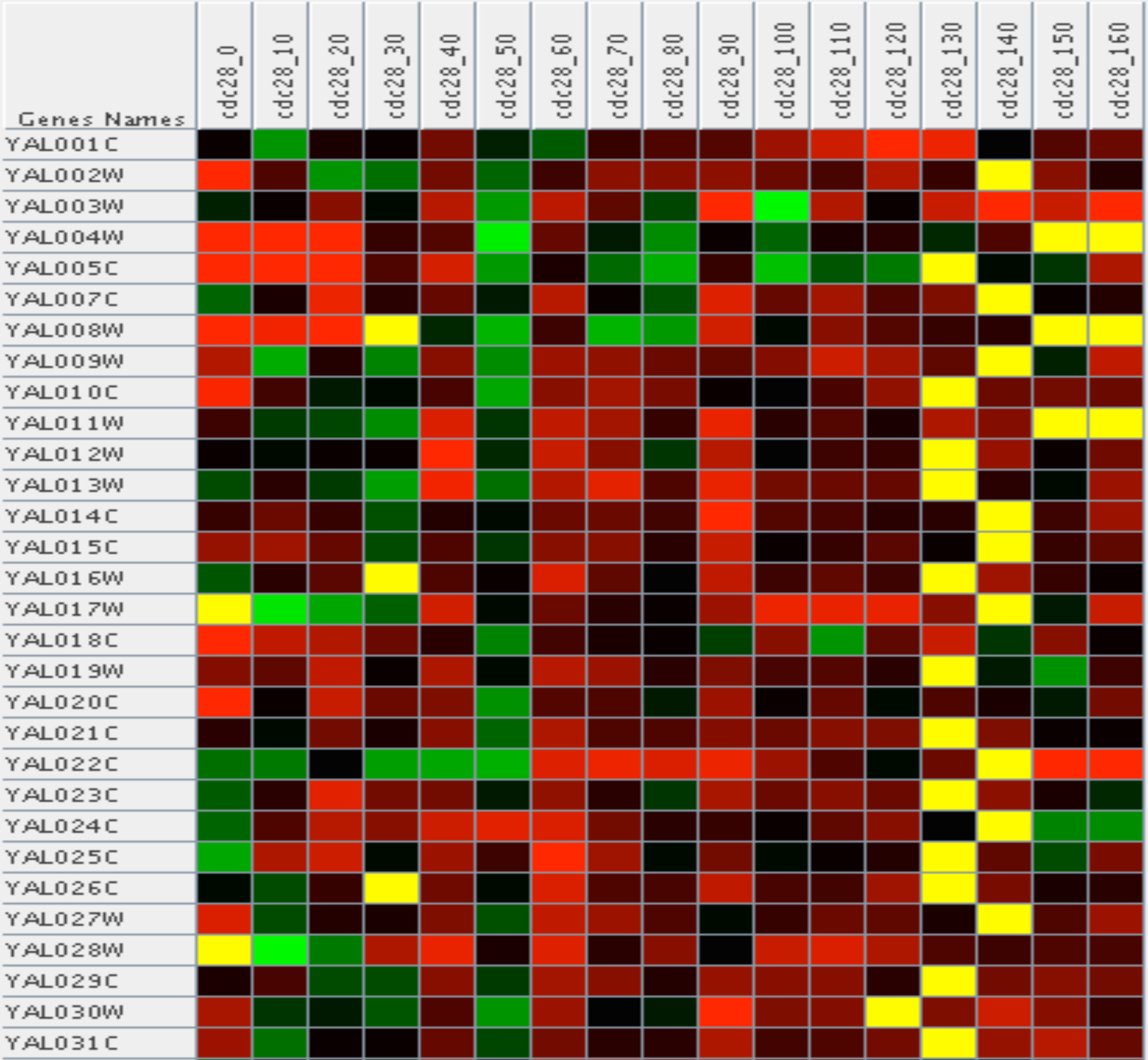
$(\{G_2, G_3, G_4\}, \{C_4, C_5, C_6\})$

Example – *Yeast* Cell Cycle



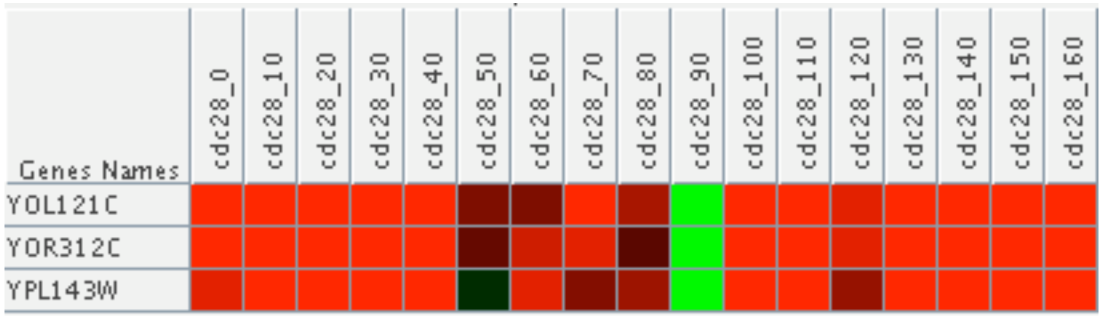
GENE_NAME	cdc28_0	cdc28_10	cdc28_20	cdc28_30	cdc28_40	cdc28_50	cdc28_60	cdc28_70	cdc28_80	cdc28_90	cdc28_100	cdc28_110	cdc28_120	cdc28_130	cdc28_140	cdc28_150	cdc28_160
YAL001C	-0.19	-0.77	-0.17	-0.19	0.13	-0.36	-0.55	-0.07	-0.01	0.03	0.27	0.49	0.85	0.66	-0.24	0.03	0.09
YAL002W	0.83	-0.01	-0.77	-0.62	0.14	-0.58	-0.05	0.23	0.2	0.23	0.08	-0.03	0.39	-0.09		0.19	-0.14
YAL003W	-0.36	-0.22	0.22	-0.28	0.41	-0.8	0.42	0.05	-0.47	1.06	-2.82	0.38	-0.22	0.47	0.89	0.48	0.8
YAL004W	1.64	1.14	0.88	-0.07	0.03	-1.18	0.07	-0.34	-0.73	-0.18	-0.6	-0.16	-0.12	-0.38	0.01		
YAL005C	1.55	1.58	1.34	0.01	0.53	-0.8	-0.16	-0.61	-0.9	-0.07	-0.96	-0.53	-0.66		-0.27	-0.41	0.35
YAL007C	-0.59	-0.16	0.66	-0.1	0.07	-0.33	0.41	-0.23	-0.51	0.58	0.07	0.32	0.01	0.17		-0.21	-0.14
YAL008W	1.15	0.67	0.94		-0.38	-0.91	-0.05	-0.91	-0.79	0.52	-0.3	0.21	0.03	-0.08	-0.1		
YAL009W	0.39	-0.87	-0.13	-0.71	0.2	-0.73	0.28	0.25	0.1	0.06	0.18	0.5	0.33	0.06		-0.35	0.44
YAL010C	0.7	-0.04	-0.33	-0.27	-0.02	-0.85	0.2	0.33	0.15	-0.18	-0.24	-0.03	0.24		0.11	0.14	0.08
YAL011W	-0.06	-0.44	-0.47	-0.73	0.54	-0.4	0.43	0.33	-0.09	0.62	-0.12	0.03	-0.17	0.36	0.18		
YAL012W	-0.18	-0.31	-0.23	-0.2	0.84	-0.37	0.47	0.19	-0.4	0.41	-0.25	-0.06	-0.09		0.26	-0.19	0.12
YAL013W	-0.49	-0.12	-0.44	-0.82	0.67	-0.62	0.38	0.6	0	0.63	0.14	0.08	0.07		-0.1	-0.26	0.28
YAL014C	-0.08	0.08	-0.07	-0.51	-0.13	-0.3	0.09	0.11	-0.04	0.87	0.02	-0.03	-0.12	-0.12		-0.05	0.28
YAL015C	0.26	0.3	0.07	-0.5	0	-0.4	0.2	0.21	-0.11	0.46	-0.19	-0.08	0.04	-0.22		-0.07	0.05
YAL016W	-0.53	-0.1	0.04		0.01	-0.19	0.54	0.05	-0.25	0.44	-0.05	0.05	-0.05		0.31	-0.08	-0.19
YAL017W		-1.15	-0.85	-0.56	0.52	-0.26	0.08	-0.11	-0.19	0.29	0.64	0.63	0.63	0.21		-0.32	0.46
YAL018C	1.24	0.43	0.38	0.09	-0.12	-0.7	-0.04	-0.15	-0.23	-0.45	0.21	-0.76	0.06	0.46	-0.4	0.21	-0.23
YAL019W	0.18	0.06	0.44	-0.23	0.36	-0.29	0.41	0.28	-0.1	0.16	-0.03	0.03	-0.12		-0.33	-0.75	-0.06
YAL020C	0.94	-0.18	0.46	0.1	0.16	-0.75	0.03	-0.01	-0.32	0.29	-0.18	0.07	-0.27	0.01	-0.16	-0.32	0.14
YAL021C	-0.12	-0.27	0.13	-0.16	0.2	-0.57	0.35	0	0	0.18	0.1	0.22	0.16		0.16	-0.19	-0.21
YAL022C	-0.63	-0.66	-0.24	-0.82	-0.84	-0.89	0.57	0.64	0.54	0.66	0.28	-0.01	-0.26	0.11		0.71	0.81
YAL023C	-0.54	-0.1	0.59	0.14	0.14	-0.33	0.24	-0.1	-0.42	0.34	0.09	0.2	0.09		0.23	-0.17	-0.39
YAL024C	-0.59	-0.01	0.41	0.2	0.5	0.59	0.54	0.14	-0.12	-0.08	-0.19	0.05	0.22	-0.24		-0.71	-0.73
YAL025C	-0.84	0.36	0.5	-0.26	0.27	-0.06	0.77	0.3	-0.26	0.13	-0.31	-0.21	-0.13		0.06	-0.48	0.15
YAL026C	-0.31	-0.5	-0.08		0.12	-0.3	0.55	0.01	-0.02	0.43	-0.03	-0.04	0.31		0.15	-0.17	-0.11
YAL027W	0.55	-0.49	-0.13	-0.17	0.17	-0.51	0.43	0.27	0.01	-0.31	-0.08	0.08	0.05	-0.16		0.01	0.28
YAL028W		-1.87	-0.65	0.35	0.63	-0.17	0.57	-0.1	0.22	-0.25	0.47	0.54	0.35	0	-0.06	0	-0.03
YAL029C	-0.15	-0.03	-0.5	-0.49	0.21	-0.43	0.33	0.2	-0.14	0.26	0.2	0.2	-0.12		0.14	0.19	0.14
YAL030W	0.34	-0.42	-0.34	-0.53	0.01	-0.76	0.29	-0.24	-0.32	0.85	0.17	0.18		0.16	0.49	0.19	-0.07
YAL031C	0.29	-0.62	-0.23	-0.19	0.07	-0.46	0.14	-0.1	-0.1	0.36	-0.04	-0.01	0.14		0.26	0.41	0.07
YAL032C	-0.48	-0.27	-0.21	-0.35	0.4	-0.18	0.41	0.47	0.19	0.39	-0.02	0.09	-0.12	-0.44	-0.25	0.16	0.22
YAL033W	-0.29	-0.04	0.32	-0.01	0.17	-0.47	0.45	0.26	-0.08	-0.06	-0.29	-0.12	-0.28		0.34	-0.2	0.29
YAL034C	0.27	-0.37	-0.18	-0.01	0.44	0.15	0.47	0.39	0.08	-0.14	-0.14	-0.04	0.21	-0.22	0.05	0.05	-1.01
YAL035C-A	-0.7	0.27	0.38	0.05	0.45	-0.12	0.25	0	-0.35	0.23	0.38	0.51	0.38	0.25	-0.28	-1.46	-0.24
YAL035W	-0.88	0.1	0.53	-0.24	0	-0.18	0.43	0.07	-0.17	0.58	-0.12	-0.07	-0.02		0.25	-0.33	0.02
YAL036C	0.01	0.04	0.66		0.18	-0.47	0.27	0.01	-0.3	0.39	-0.35	-0.15	-0.14	0.12		-0.28	0.01
YAL037W	1.11	-0.13	0.56	0.02	0.08	-0.47	0.16	-0.18	-0.31	-0.92	0.08	0.16	0.25	-0.11	0	-0.06	-0.25
YAL038W	0.23	-0.42	0.35	-0.06	0.65	-0.68	0.28	0.05	-0.45		-2.69	0.49	-0.17	0.41	0.76	0.5	0.74
YAL039C	0.83	0.1	-0.31	-0.49	0.06	-0.67	-0.01	0	-0.1	0.68	0.33	0.13	-0.29		-0.23	-0.33	0.33
YAL040C	-0.05	-0.15	-0.58	-0.58	0.18	-0.67	0.29	0.2	0.02	1.32	0.27		-0.41	-0.05	0.17	0.14	-0.11
YAL041W	0.03	-0.32	-0.49	-0.36	0.4	-0.35	0.34	0.28	0.02	0.55	-0.09	-0.13	0.2		0.19	-0.08	-0.21
YAL042W	0	-0.17	0.21	-0.2	0.26	-0.73	-0.04	0.21	-0.2	0.75	-0.14	0.06	-0.04	-0.21		0.05	0.2
YAL043C	0.09	-0.35	-0.13	-0.25	0.32	-0.48	-0.08	0.4	0.12	0.73	0.08	-0.15	-0.11		-0.02	-0.25	0.09
YAL043C-A	0.09	-0.01	-0.28	-0.18	0.2	-0.99	0.2	0.51	0.17	0.14	0.03	0.21	0.07	-0.09		-0.23	0.16
YAL044C	-0.13	-0.28	-0.08	-0.64	-0.82	-1.01	-0.24	-0.32	-0.28	1.03	0.31	0.48	-0.15		0.86	0.2	1.07
YAL045C	-0.1	0.12	0.32	-0.33	-0.11	-0.31	-0.14	0.08	0.06	0.6	0.14	0.12	-0.09	-0.26		-0.19	0.09
YAL046C	0.11	0.14	0.13	-0.5	0.01	-0.18	0.14	0.08	-0.18	0.68	0	0.02	-0.21	-0.26		-0.09	0.11
YAL047C	-0.26	0.17	0.46	-0.19	-0.14	-0.35	0.1	-0.22	-0.15	0.37	-0.05	-0.02	0	0.19		0.18	-0.09

Example – *Yeast* Cell Cycle

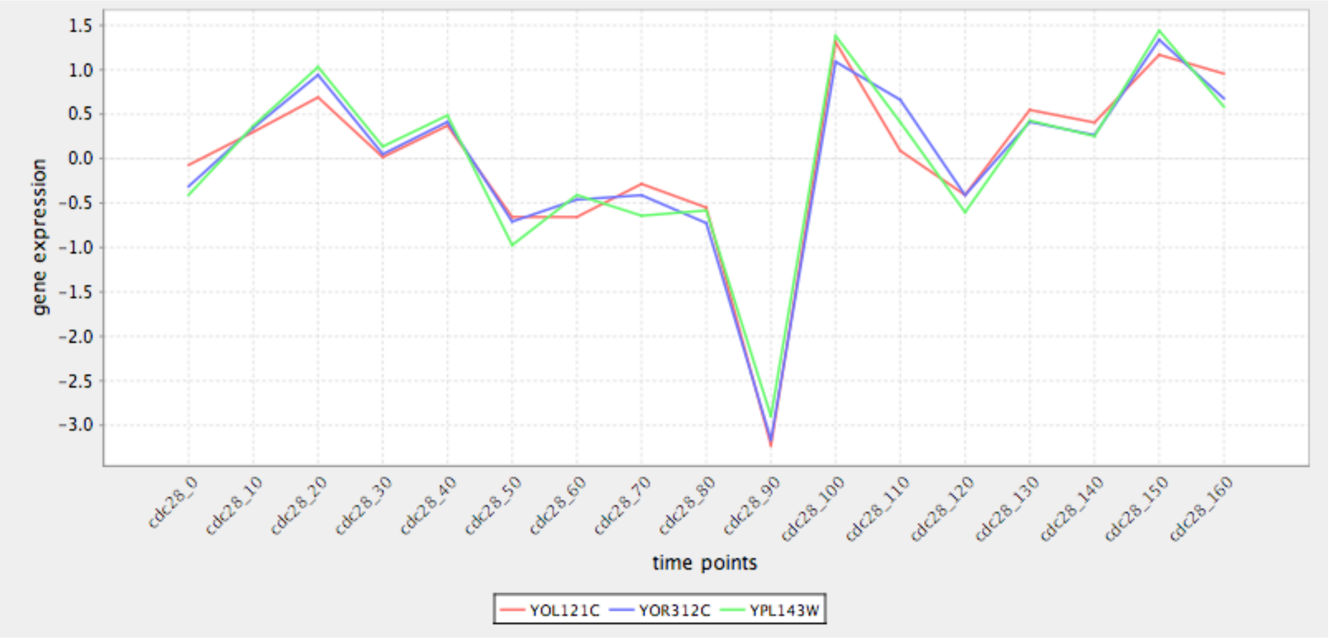


Example – *Yeast* Cell Cycle

GENE CLUSTER

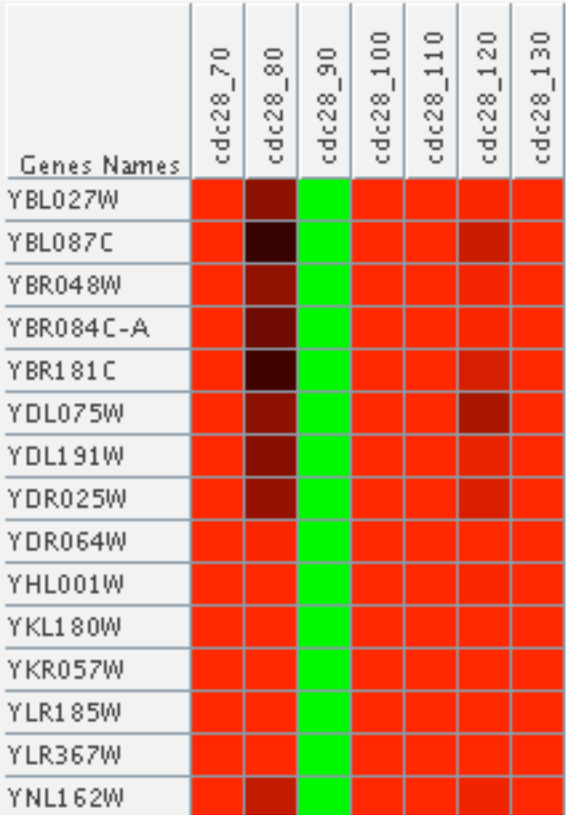


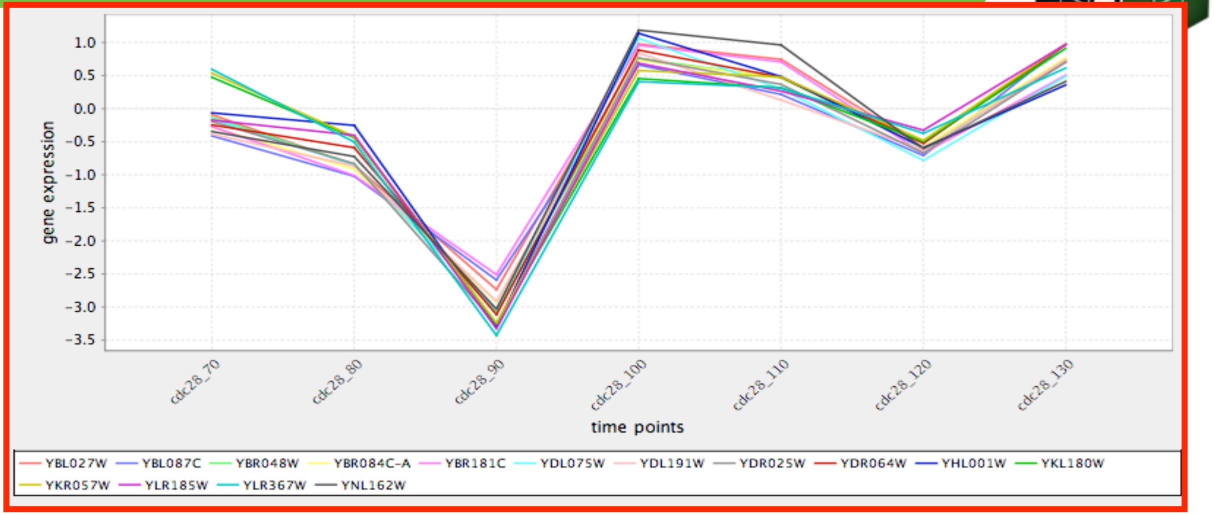
Genes Names	cdc28_0	cdc28_10	cdc28_20	cdc28_30	cdc28_40	cdc28_50	cdc28_60	cdc28_70	cdc28_80	cdc28_90	cdc28_100	cdc28_110	cdc28_120	cdc28_130	cdc28_140	cdc28_150	cdc28_160
YOL121C	🟥	🟥	🟥	🟥	🟥	🟫	🟫	🟥	🟫	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥
YOR312C	🟥	🟥	🟥	🟥	🟥	🟫	🟥	🟥	🟫	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥
YPL143W	🟥	🟥	🟥	🟥	🟥	⬛	🟥	🟫	🟫	🟩	🟥	🟥	🟫	🟥	🟥	🟥	🟥

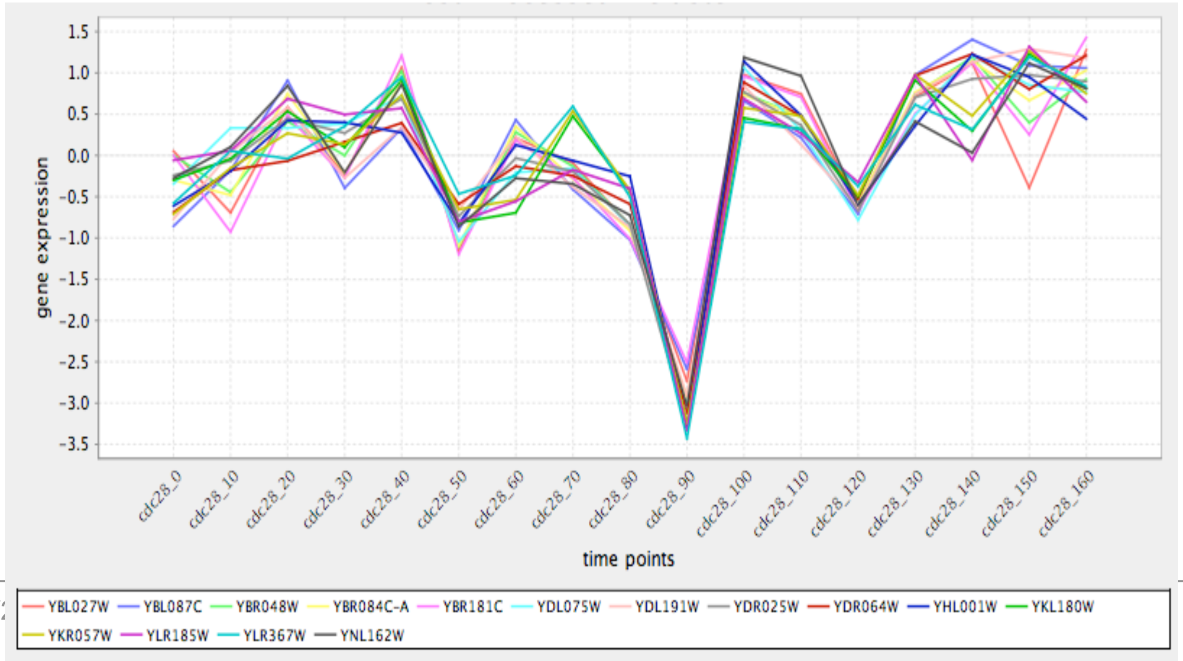


Example – *Yeast* Cell Cycle

BICLUSTER

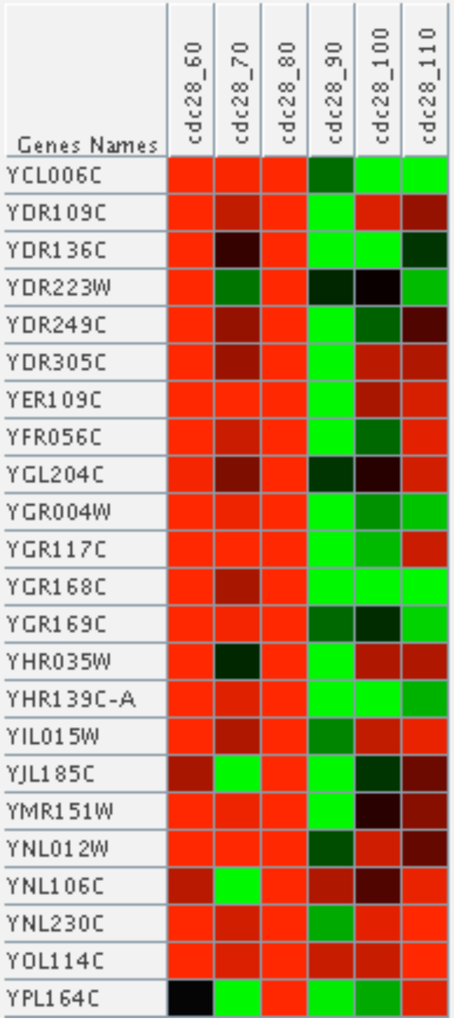


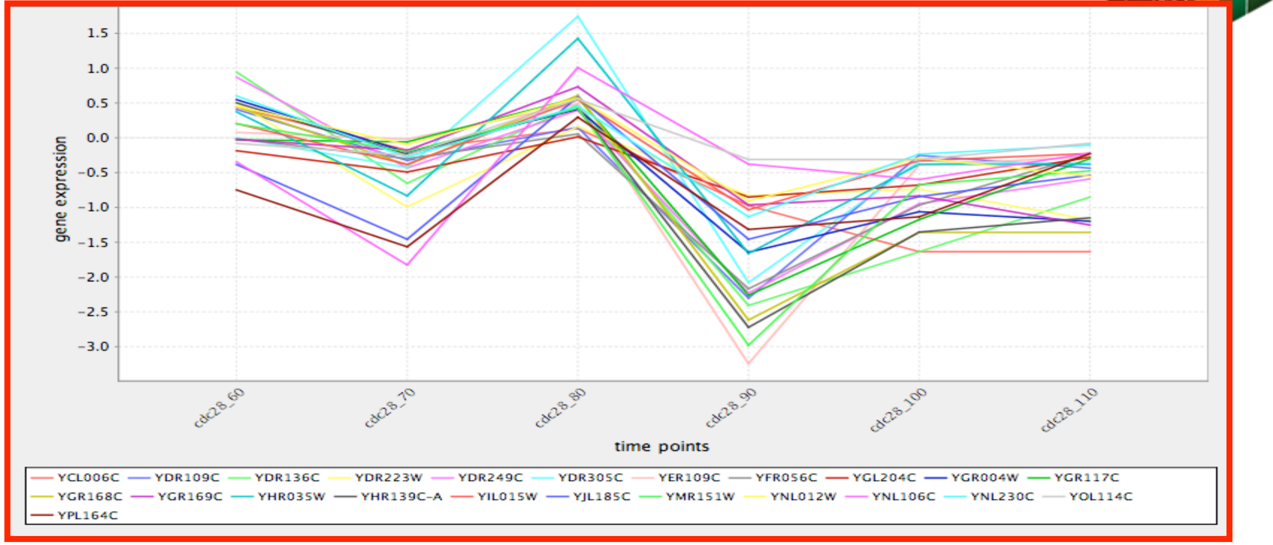


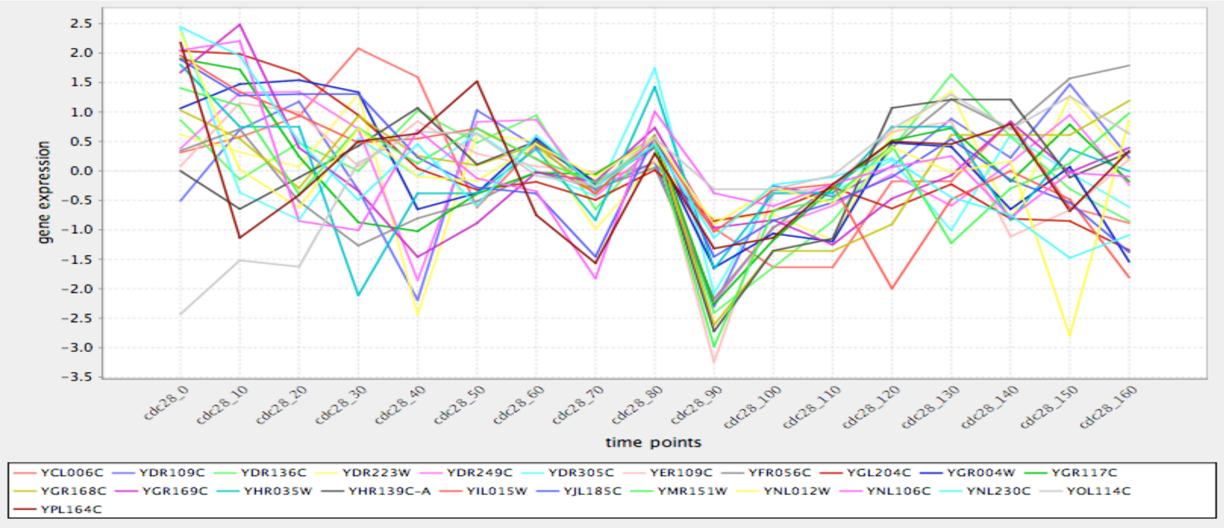


Example – *Yeast* Cell Cycle

BICLUSTER







Why Biclustering and not just Clustering?

- When Clustering algorithms are used

Global Model

  - Each gene in a given gene cluster is defined using all the conditions.
  - Each condition in a condition cluster is characterized by the activity of all the genes.

- When Biclustering algorithms are used

Local Model

  - Each gene in a bicluster is selected using only a subset of the conditions
  - Each condition in a bicluster is selected using only a subset of the genes.

Why Biclustering and not just Clustering?

- Unlike Clustering
  - Biclustering identifies groups of genes that show similar activity patterns under a specific subset of the experimental conditions.
- Biclustering is the key technique to use when
  1. Only a small set of the genes participates in a cellular process of interest.
  2. An interesting cellular process is active only in a subset of the conditions.
  3. A single gene may participate in multiple pathways that may or not be co-active under all conditions.

Roadmap

- Clustering
- **Biclustering**
  - Why Biclustering and not just Clustering?
  - **Bicluster Types and Structure**
  - Algorithms
- Biclustering Gene Expression Time Series

Bicluster Types

1. Biclusters with constant values.

2. Biclusters with constant values on rows or columns.

3. Biclusters with coherent values.

4. Biclusters with coherent evolutions.

Constant Values

- **Perfect constant bicluster**

sub-matrix *(I,J)* where all values within the bicluster are equal for all $i \in I$ and $j \in J$:

$$a_{ij} = \mu$$

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

Constant Values on Rows or Columns

- **Perfect bicluster with constant rows**
  - submatrix *(I,J)* where all the values within the bicluster can be obtained using:

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

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

where μ is the typical value within the bicluster and α_i is the adjustment for row $i \in I$.

- **Perfect bicluster with constant columns**
  - submatrix *(I,J)* where all the values within the bicluster can be obtained using:

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

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

where μ is the typical value within the bicluster and β_j is the adjustment for column $j \in J$.

This adjustment can be obtained either in an additive or multiplicative way.

Constant Values on Rows or Columns

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

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

Constant Columns

Coherent Values

- **Perfect bicluster with additive/multiplicative model**
  - a subset of rows and a subset of columns, whose values a_{ij} are predicted using:

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

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

where μ is the typical value within the bicluster, α_i is the adjustment for row $i \in I$ and β_j is the adjustment for row $j \in J$.

These adjustments can be obtained either in an additive or multiplicative way.

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

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

Multiplicative Model

Coherent Values

- The **"the plaid models"** (Lazzeroni and Owen) consider a generalization of the additive model: **general additive model**.

- For every element a_{ij}
  - The general additive model represents a sum of models.
  - Each model represents the contribution of the bicluster B_k to the value of a_{ij} in case $i \in I$ and $j \in J$.

Coherent Values

General Additive Model

$$a_{ij} = \sum_{k=0}^{K} \theta_{ijk} \rho_{ik} \kappa_{jk}$$

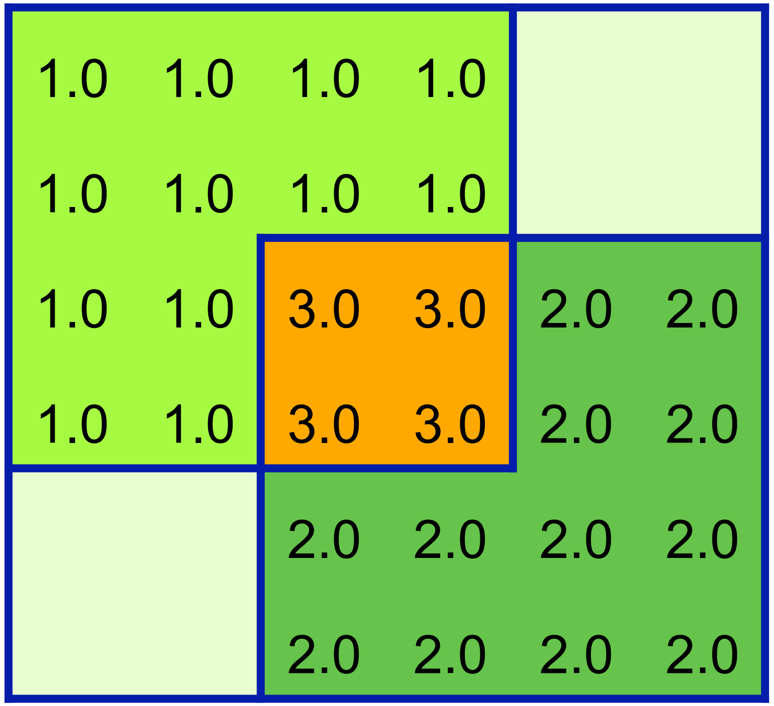
- K is the number of biclusters.
- ρ_{ik} and κ_{jk} are binary values that represent memberships:
  - ρ_{ik} *is* the membership of row *i* in the bicluster *k.*
  - κ_{jk} is the membership of column *j* in the bicluster *k.*

- θ_{ijk} specifies the contribution of each bicluster *k* and can be one of the following expressions representing different types of biclusters:
  - μ_k → Constant Biclusters
  - $\mu_k + \alpha_{ik}$ → Biclusters with constant rows
  - $\mu_k + \beta_{jk}$ → Biclusters with constant columns
  - $\mu_k + \alpha_{ik} + \beta_{jk}$ → Biclusters with additive model

General Multiplicative Model can also be assumed!

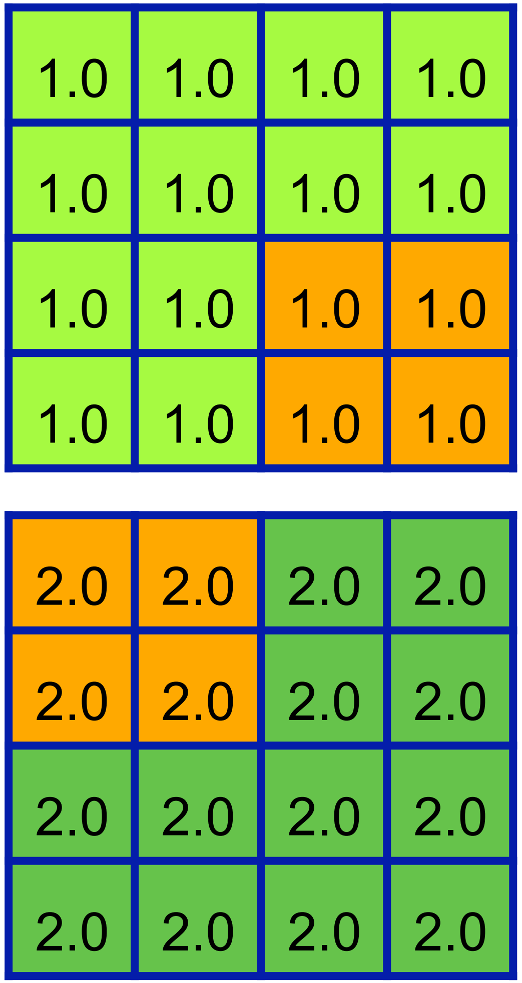
Coherent Values

General Additive Model



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

Constant Biclusters



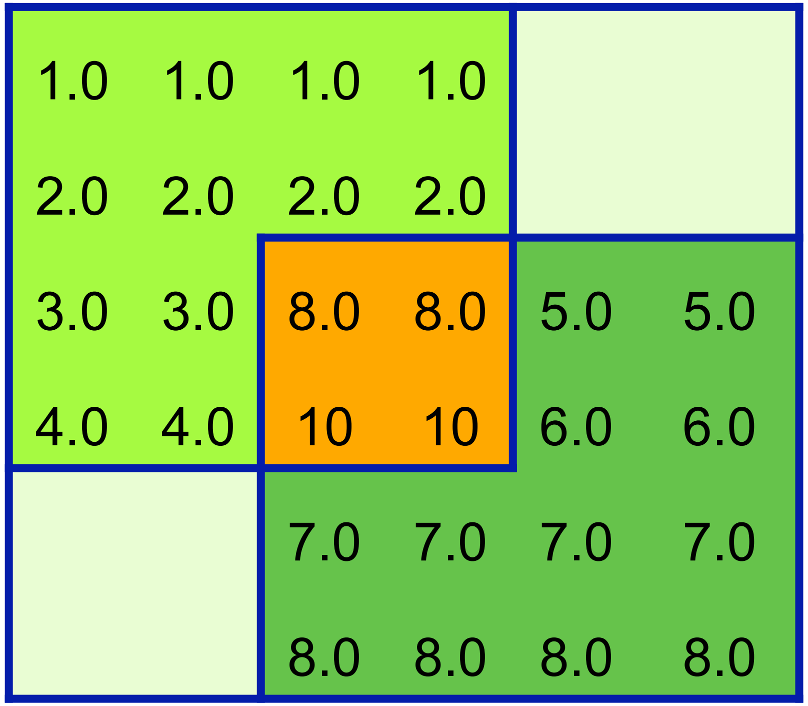
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0

2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0

Constant Values

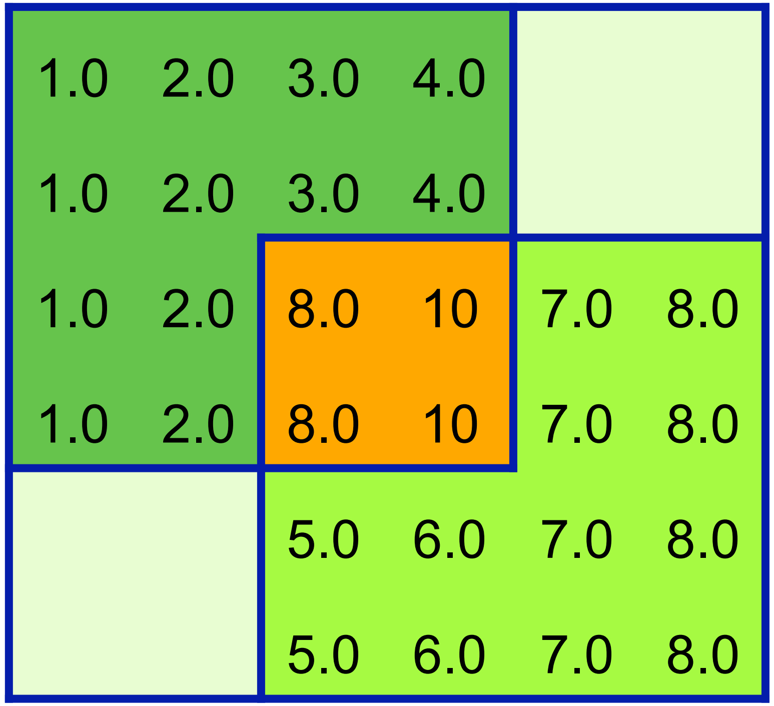
Coherent Values

General Additive Model



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	10	6.0	6.0
		7.0	7.0	7.0	7.0
		8.0	8.0	8.0	8.0

Constant Rows

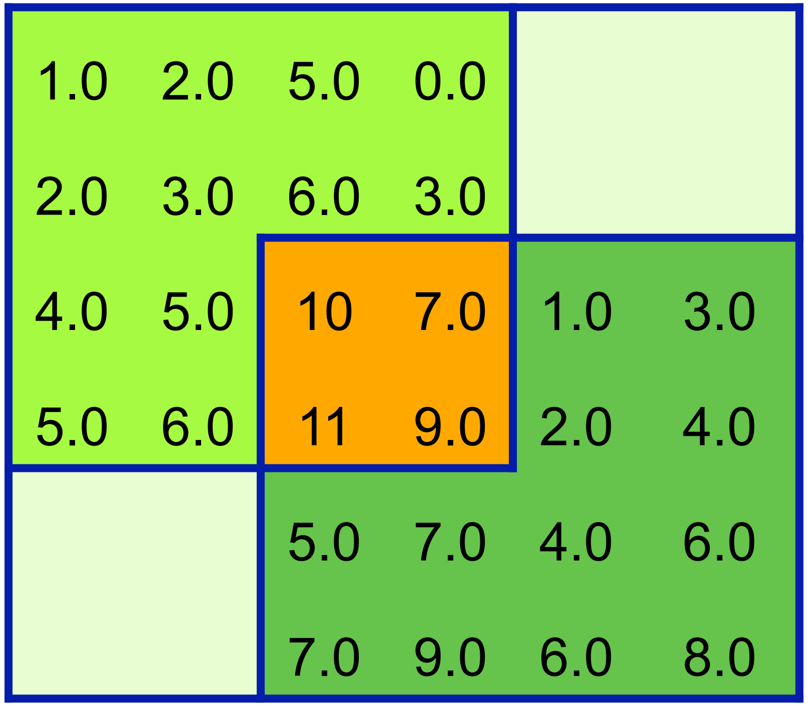


1.0	2.0	3.0	4.0		
1.0	2.0	3.0	4.0		
1.0	2.0	8.0	10	7.0	8.0
1.0	2.0	8.0	10	7.0	8.0
		5.0	6.0	7.0	8.0
		5.0	6.0	7.0	8.0

Constant Columns

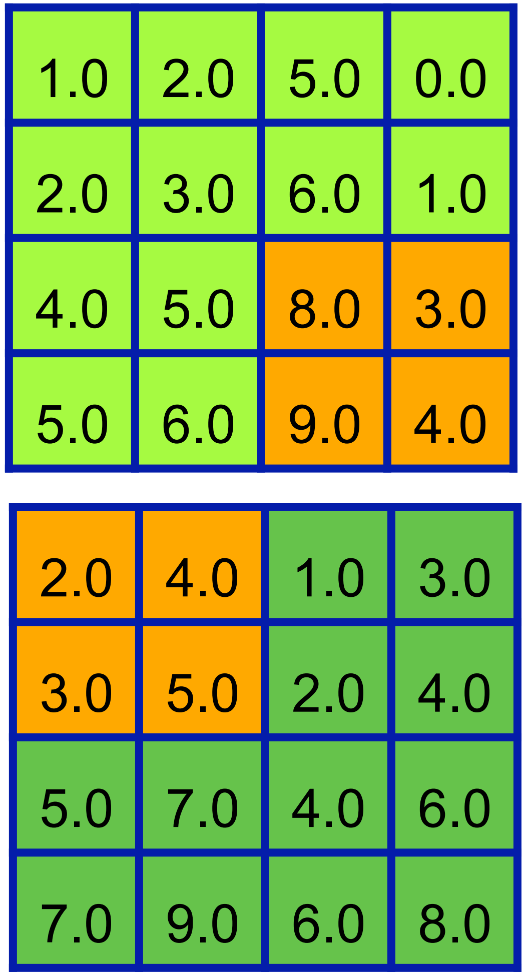
Coherent Values

General Additive Model



1.0	2.0	5.0	0.0		
2.0	3.0	6.0	3.0		
4.0	5.0	10	7.0	1.0	3.0
5.0	6.0	11	9.0	2.0	4.0
		5.0	7.0	4.0	6.0
		7.0	9.0	6.0	8.0

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

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

Additive Model

Coherent Evolutions

- Elements of the matrix are viewed as symbolic values.
- Try to discover biclusters with coherent behaviors regardless of the exact numeric values in the data matrix.
- The co-evolution property can be observed:
  - On the entire bicluster
  - On the rows of the bicluster
  - On the columns of the bicluster

Coherent Evolutions

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

Overall Coherent

Evolution

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

Coherent Evolution

On the Rows

Coherent Evolutions

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

Coherent Evolution

On the Columns

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

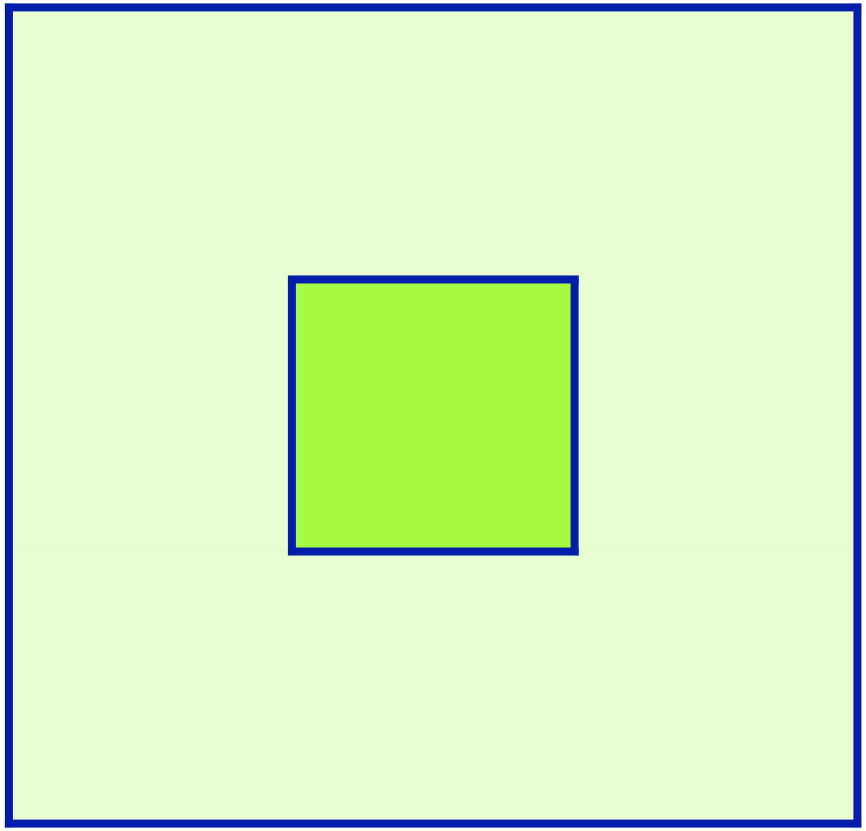
Order Preserving

Sub-Matrix (OPSM)

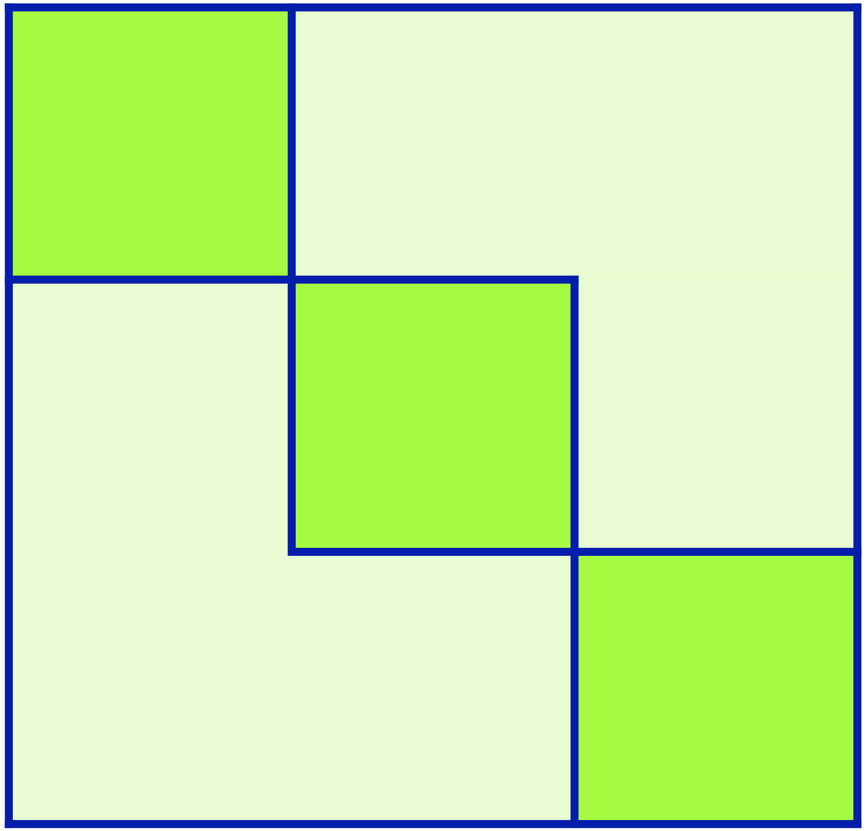
Biclustering Structure

- **One Bicluster**
- **Several Biclusters**
  - Exclusive-Rows Biclusters
  - Exclusive-Columns Biclusters
  - Non-Overlapping Biclusters with Tree Structure
  - Non-Overlapping Non-Exclusive Biclusters
  - Overlapping Biclusters with Hierarchical Structure
  - Arbitrarily Positioned Overlapping Biclusters

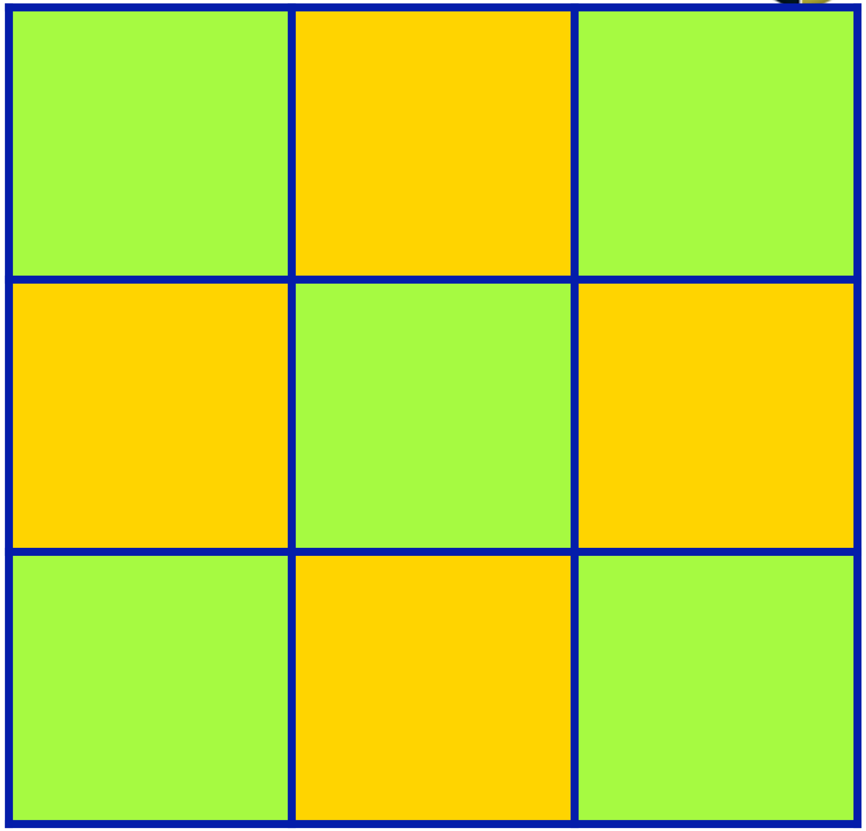
One Bicluster



Several Biclusters

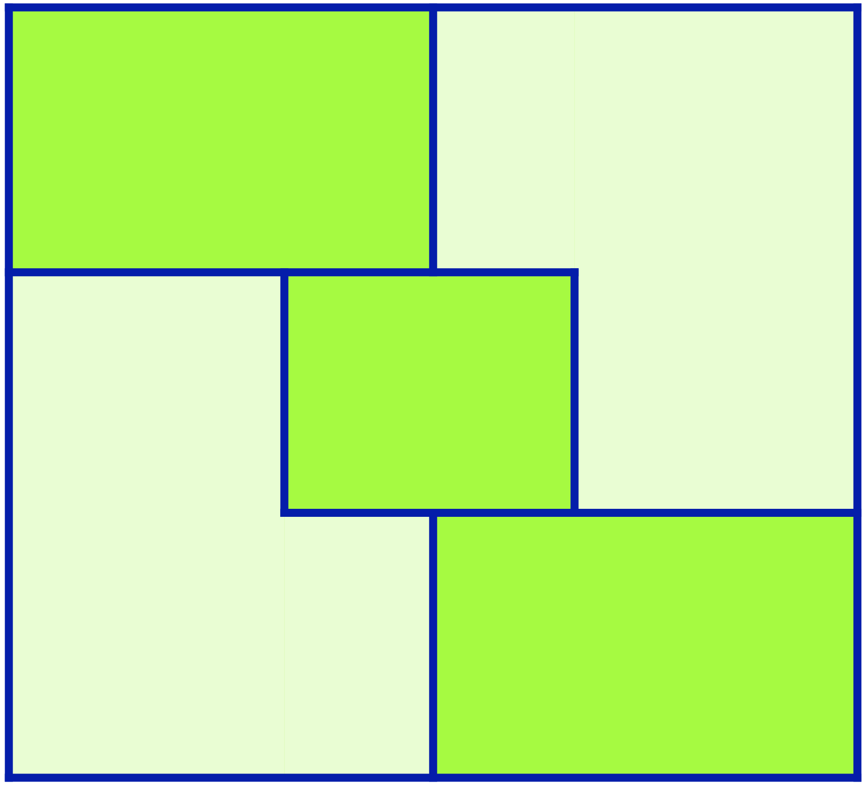


Exclusive Row and Column Biclusters

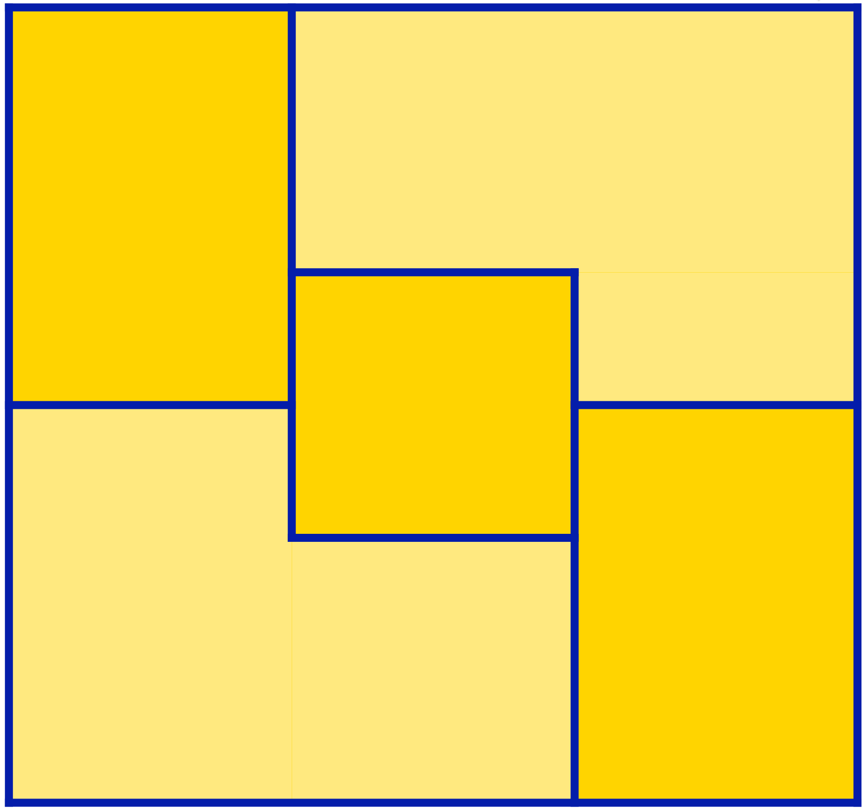


Checkerboard Structure

Several Biclusters

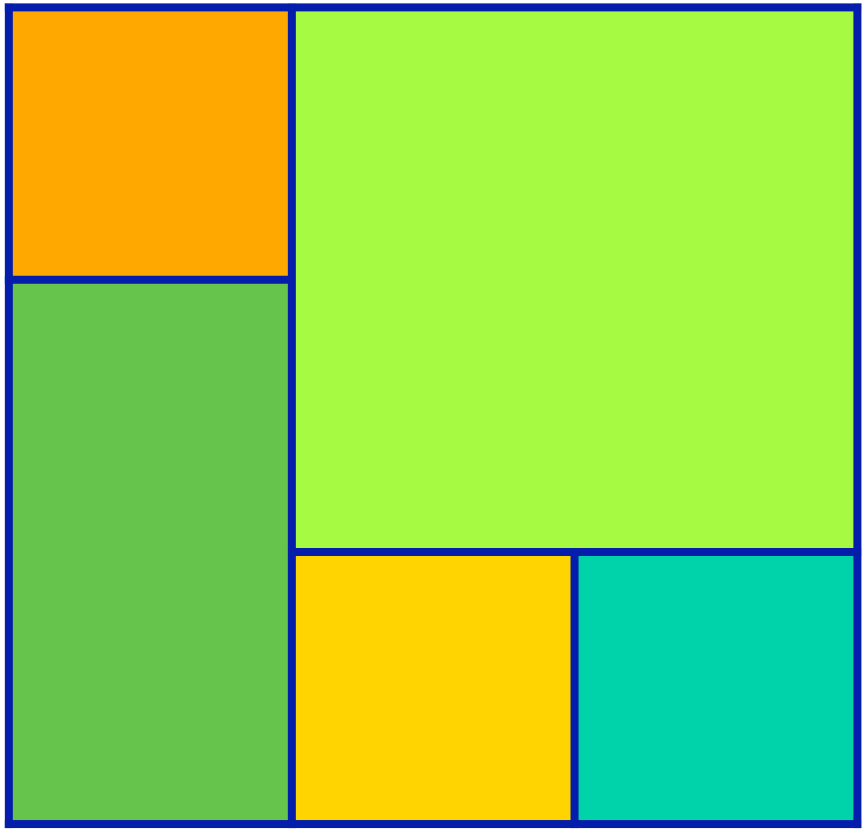


Exclusive-Rows Biclusters

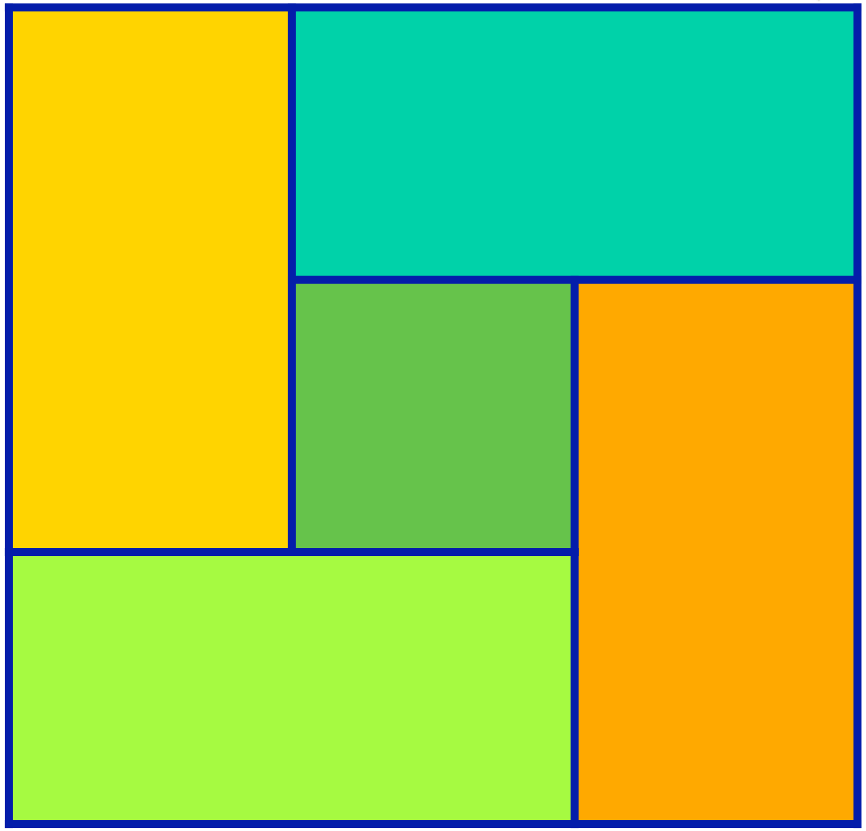


Exclusive-Columns Biclusters

Several Biclusters

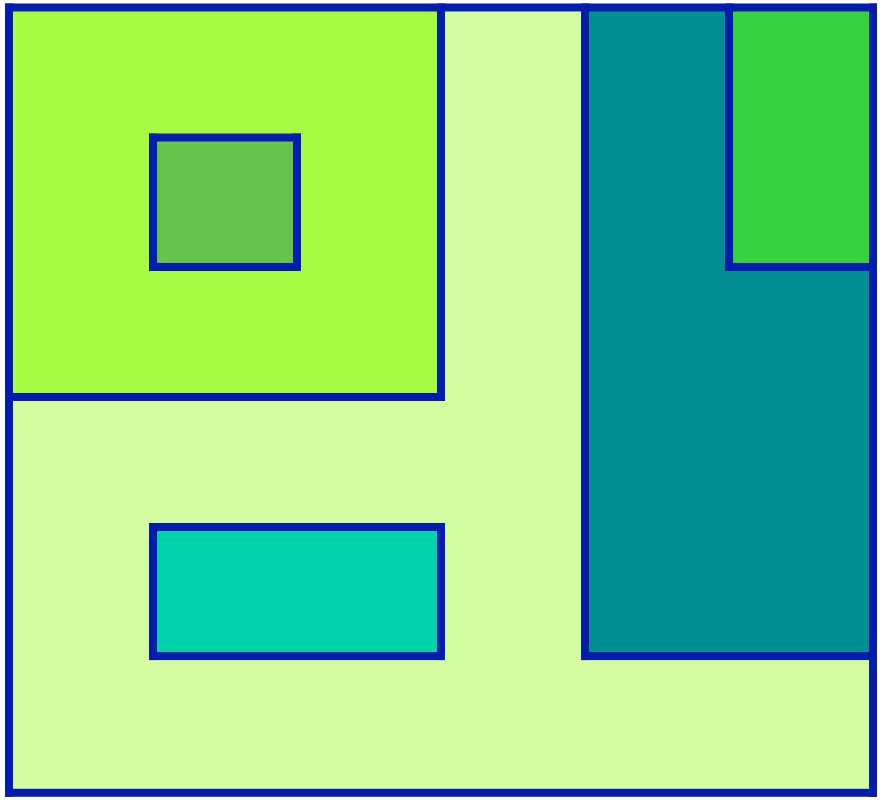


Non-Overlapping Biclusters with Tree Structure

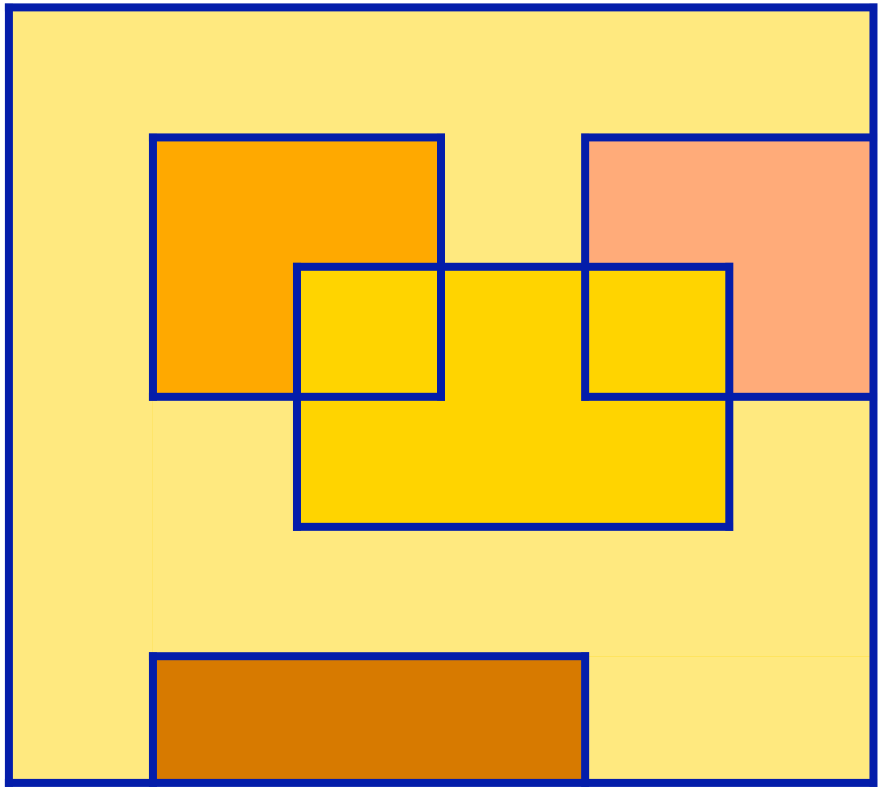


Non-Overlapping Non-Exclusive Biclusters

Several Biclusters



Overlapping Biclusters with Hierarchical Structure



Arbitrarily Positioned Overlapping Biclusters

Roadmap

- Clustering
- **Biclustering**
  - Why Biclustering and not just Clustering?
  - Bicluster Types and Structure
  - **Algorithms**
- Biclustering Gene Expression Time Series

Biclustering Algorithms

- **Different Goals**
  - Identify one bicluster.
  - Identify a given number of biclusters.
- **Different Approaches**
  - Discover one bicluster at a time.
  - Discover one set of biclusters at a time.
  - Discover all biclusters at the same time (Simultaneous bicluster identification)

Biclustering Algorithms

- **Iterative Row and Column Clustering Combination**
  - Apply clustering algorithms to the rows and columns of the data matrix, separately.
  - Use an iterative procedure to combine the two clustering results.
- **Divide and Conquer**
  - Break the problem into several subproblems similar to the original problem but smaller in size.
  - Solve the subproblems recursively.
  - Combine the intermediate solutions to create a solution to the original problem.
  - Usually break the matrix into submatrices (biclusters) based on a certain criterion and then continue the biclustering process on the new submatrices.

Biclustering Algorithms

- **Greedy Iterative Search**
  - Always make a locally optimal choice in the hope that this choice will lead to a globally good solution.
  - Usually perform greedy row/column addition/removal.
- **Exhaustive Bicluster Enumeration**
  - A number of methods have been used to speed up exhaustive search.
  - In some cases the algorithms assume restrictions on the size of the biclusters that should be listed.

State of the Art

- CC (Cheng and Church, ISMB 2000)
- Plaid models (Lazzeroni and Owen, Statistica Sinica 2002)
- SAMBA (Tanay et al, Bioinformatics 2002)
- OPSM (Ben-Dor et al, JCB 2003)
- X-Motifs (Murali and Kasif, PCB 2003)
- ISA - *Iterative Signature Algorithm* (Ihmels et al, Bioinformatics 2004)
- BiMax (Prelic et al, Bioinformatics 2006)
- BiMine (Ayadi et al*, BioData Mining 2009)*
- QUBIC (Li et al, NAR 2009)
- FABIA (Hochreiter, Bioinformatics 2010)
- ...

Roadmap

- Biclustering
  - Why Biclustering and not just Clustering?
  - Bicluster Types and Structure
  - Algorithms
- **Biclustering Gene Expression Time Series**
  - **Context and Motivation**
    - **Importance of Expression Time Series, Problem restriction, and biclusters with contiguous columns**
  - **State of the art**
  - CCC-Biclustering algorithm

Context and Motivation

- **Time series gene expression data** enable
  - Study gene expression over time (dynamics)
  - Discovery of coherent temporal expression patterns
- **Critical to understand complex biomedical problems**
  - Development
  - Response to stress
  - Disease progression
  - Drug response
  - …

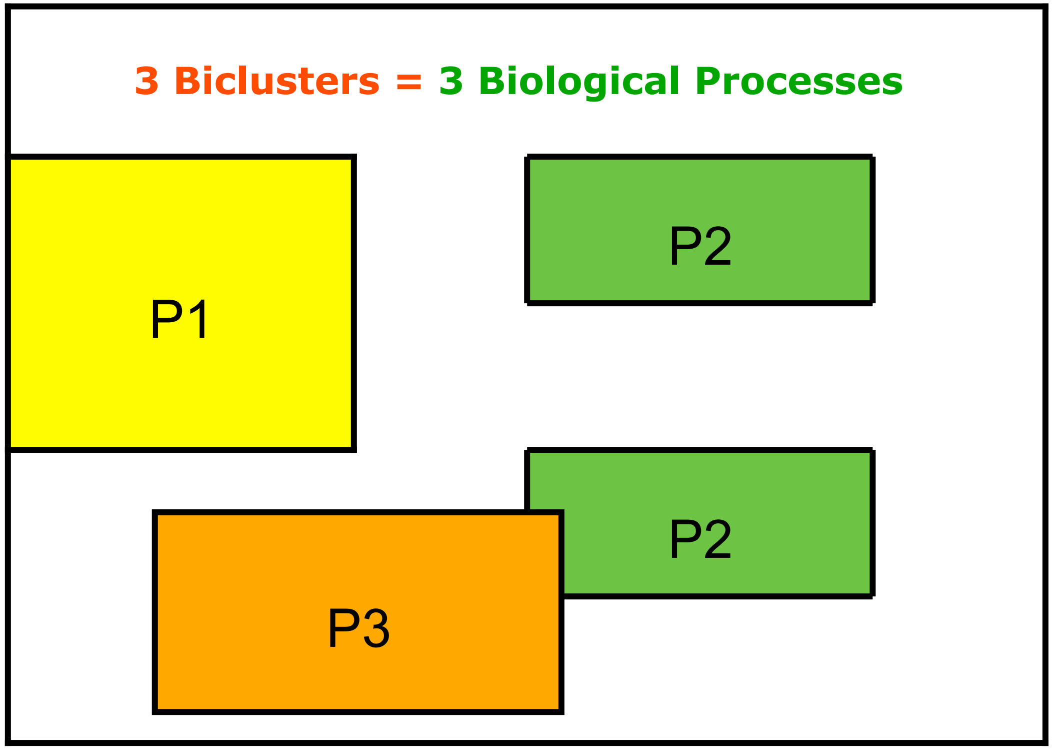
Context and Motivation

- **Biclustering** recognized as effective method
  - Discover **local expression patterns**.
  - Unravel potential regulatory mechanisms.
- **Most biclustering formulations are NP-hard**.
  - Many algorithms for gene expression in general (suboptimal results in time series).
- **Few algorithms for special case of time series** (not efficient computational or biologically).
- **Need for specific and efficient biclustering algorithms to analyze expression time series !!**

Context and Motivation

- **Problem Restriction**
  - When analyzing gene expression time series, the biclustering problem can be restricted to the identification of **biclusters with contiguous columns**.
- Restriction is biologically reasonable.
- Leads to a **tractable problem** and efficient biclustering algorithms.
- **Biological Assumption**
  - The activation of a set of genes under specific conditions corresponds to the activation of a particular biological process.
  - As time goes on, **biological processes** start and finish, leading to increased (or decreased) activity of genes, that can be identified since they form **biclusters with contiguous columns**.

Context and Motivation

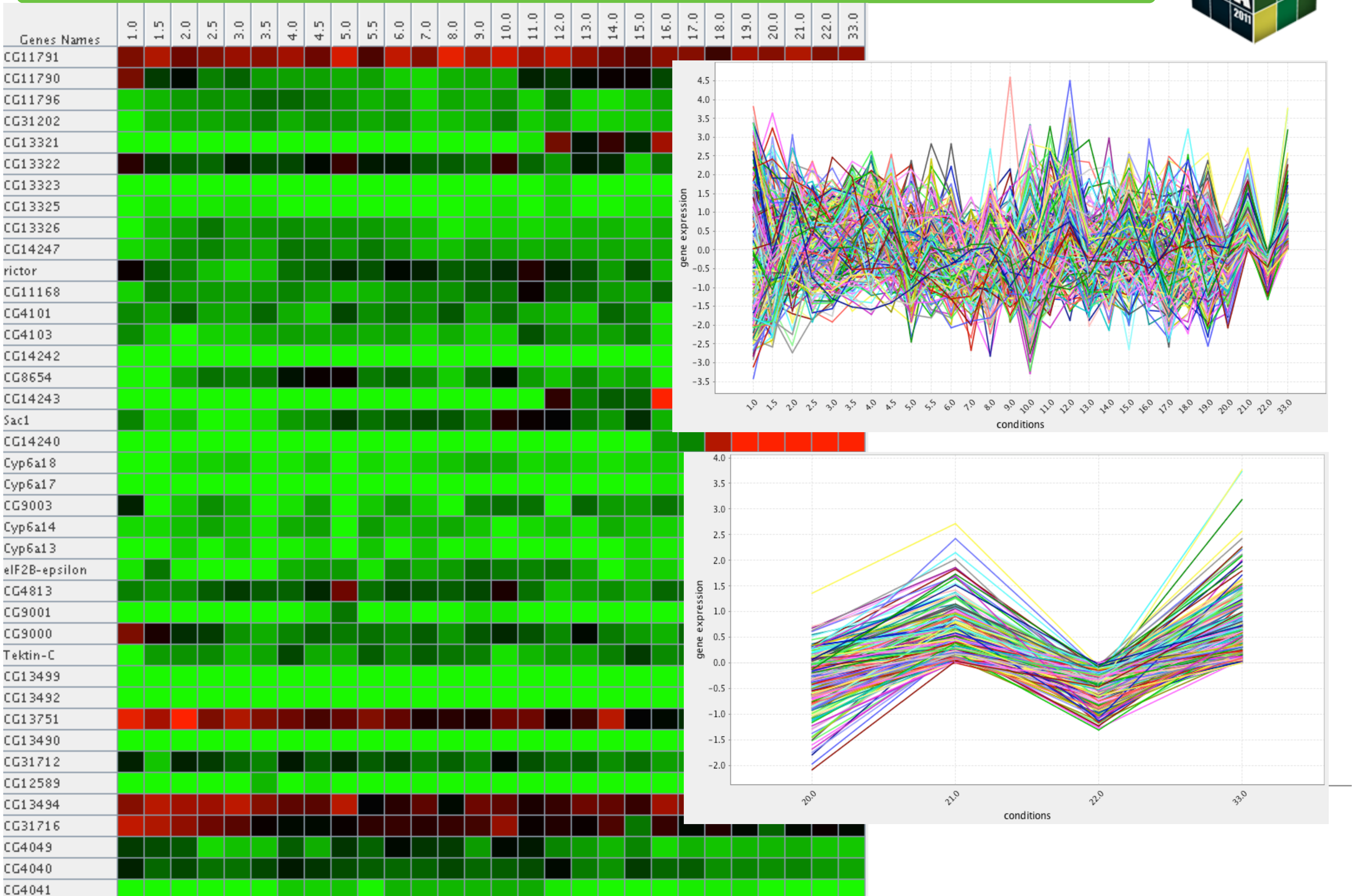


GOAL

Find biclusters with contiguous columns

(not biclusters with any set of columns!)

Context and Motivation



State of the Art

- CC-TSB Algorithm (Zhang et al., IEEE ITCC 2005)
- Q-clustering (Ji and Tan, Bioinformatics 2005)
- EDISA (Supper et al., BMC Bioinformatics 2007)
- *e*-CCC-Biclustering (Madeira and Oliveira, APBC 2007, AMB 2009)
- **CCC-Biclustering (Madeira et al., WABI 2005, IEEE/ACM TCBB 2010)**
- **...**

Roadmap

- Biclustering
  - Why Biclustering and not just Clustering?
  - Bicluster Types and Structure
  - Algorithms
- **Biclustering Gene Expression Time Series**
  - Context and Motivation
    - Importance of Expression Time Series, Problem restriction, and biclusters with contiguous columns
  - State of the art
  - **CCC-Biclustering algorithm**

Discretizing Time Series Gene Expression Data

Case of Interest: gene expression levels can be *discretized* to a set of symbols ∑ (set of distinct activation levels)

- ∑ = {D, N, U} = {Down-Regulated, No-Change, Up-Regulated}

Matrix A'	C1	C2	C3	C4	C5
Gene 1	0.07	0.73	-0.54	0.45	0.25
Gene 2	-0.34	0.46	-0.38	0.76	-0.44
Gene 3	0.22	0.17	-0.11	0.44	-0.11
Gene 4	0.70	0.71	-0.41	0.33	0.35

Gene Expression Matrix

Matrix A	C1	C2	C3	C4	C5
Gene 1	N	U	D	U	N
Gene 2	D	U	D	U	D
Gene 3	N	N	N	U	N
Gene 4	U	U	D	U	U

Discretized Expression Matrix

CCC-Biclusters

- A **Bicluster** is a subset of rows $I = \{i_1,\ldots,i_k\}$ and a subset of columns $J=\{j_1,\ldots,j_s\}$ from matrix A, such that it can be defined as a ***k* by *s* sub-matrix of matrix *A***.

- A **Trivial Bicluster** is a Bicluster with only one row or only one column.

- A **CC-Bicluster** (**Coherent Column Bicluster**) is a subset of rows $I = \{i_1,\ldots,i_k\}$ and a subset of columns $J=\{j_1,\ldots,j_l\}$ from matrix A such that $\mathbf{A_{ij} = A_{lj}}$ **, for all** $\boldsymbol{i \in I}$ **and** $\boldsymbol{j \in J}$ ***(constant columns).***

- A **CCC-Bicluster** (**Contiguous Column Coherent Bicluster**) is a subset of rows $I = \{i_1,\ldots,i_k\}$ and a **contiguous** subset of columns $J=\{j_r, j_{r+1} \ldots, j_{s-1,} j_s\}$ from matrix A such that $A_{ij} = A_{lj}$ for all $i \in I$ and $j \in J$ ***(contiguous constant columns)***.

Each CCC-Bicluster defines a **string *S*** that corresponds to an **Expression Pattern** common to every row in the CCC-Bicluster (between columns *r* and *s* of matrix A).

Maximal CCC-Biclusters

- A **CCC-Bicluster is Row-Maximal** if no more rows can be added to its set of rows *I* while maintaining the coherence property.

- A **CCC-Bicluster is Right-Maximal** if its expression pattern *S* cannot be extended to the right by adding one more symbol at its end (the column contiguous to its last column of cannot be added to *J* without removing genes from *I*).

- A **CCC-Bicluster is Left-Maximal** if its expression pattern *S* cannot be extended to the left by adding one more symbol at its beginning (the column contiguous to its first column of cannot be added to *J* without removing genes from *I*).

- A **CCC-Bicluster is Maximal** if it is Row-Maximal, Left-Maximal and Right-Maximal.

→ **NO** other CCC-Bicluster exists that properly contains it, that is, if for all other CCC-biclusters (L,M), $I \subseteq L$ *and* $J \subseteq M \Rightarrow I = L \wedge J = M$.

Maximal Non-Trivial CCC-Biclusters

Each CCC-Bicluster defines a String corresponding to an Expression Pattern common to every row in the CCC-Bicluster.

Matrix A'	C1	C2	C3	C4	C5
Gene 1	0.07	0.73	-0.54	0.45	0.25
Gene 2	-0.34	0.46	-0.38	0.76	-0.44
Gene 3	0.22	0.17	-0.11	0.44	-0.11
Gene 4	0.70	0.71	-0.41	0.33	0.35

Matrix A	C1	C2	C3	C4	C5
Gene 1	N	U	D	U	N
Gene 2	D	U	D	U	D
Gene 3	N	N	N	U	N
Gene 4	U	U	D	U	U

B1 =({G1,G2,G4},{C2,C3,C4}, [UDU])

B2 = ({G1,G3},{C4,C5}, [UN])

After Alphabet Transformation …

Each CCC-Bicluster defines a String corresponding to an Expression Pattern common to every row in the CCC-Bicluster.

Matrix A	C1	C2	C3	C4	C5
Gene 1	N	U	D	U	N
Gene 2	D	U	D	U	D
Gene 3	N	N	N	U	N
Gene 4	U	U	D	U	U

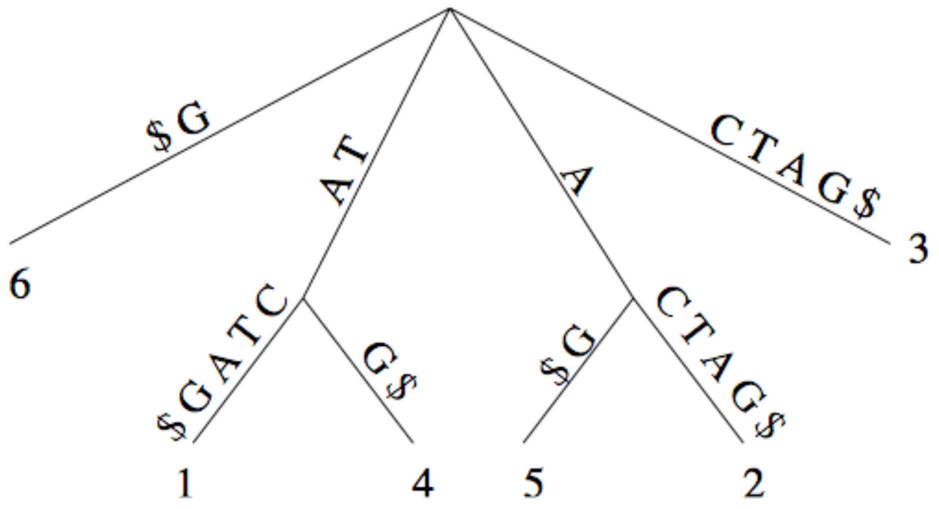
Matrix A	C1	C2	C3	C4	C5
Gene 1	N1	U2	D3	U4	N5
Gene 2	D1	U2	D3	U4	D5
Gene 3	N1	N2	N3	U4	N5
Gene 4	U1	U2	D3	U4	U5

B1 =({G1,G2,G4},{C2,C3,C4}, **[UDU]**)

B2 = ({G1,G3},{C4,C5}, **[UN]**)

Suffix Trees

- A **suffix tree** of a |*S*|-character string *S* is a rooted directed tree with exactly |S| leaves, numbered 1 to |*S*|.
- S[i…|S|] is the **suffix** of S that starts at position *i* and end at position |S|, where |*S*| is the number of characters in the string.



TACTAG

ACTAG

CTAG

TAG

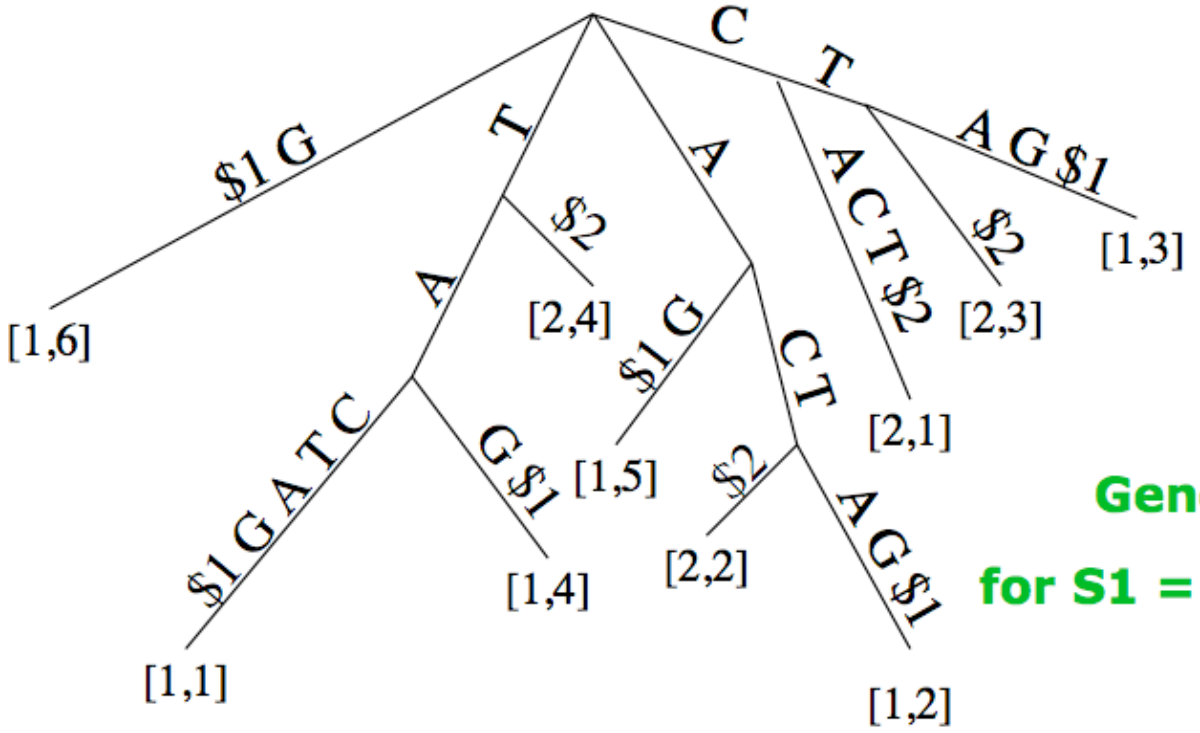
AG

G

Suffix Tree for S = TACTAG

Generalized Suffix Tree

- A **generalized suffix tree** is a suffix tree built for a set of strings $S = \{S_1, \ldots S_k\}$.



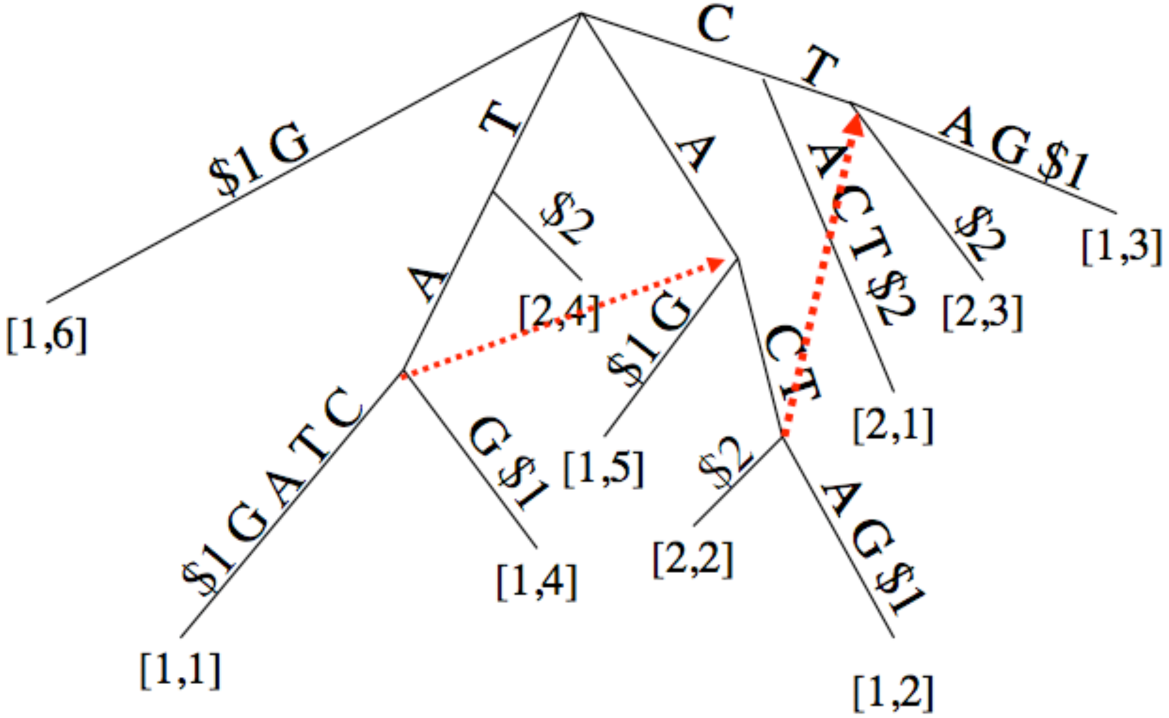
Generalized Suffix Tree

for S1 = TACTAG and S2 = CACT

- A suffix tree/generalized suffix tree can be built in **linear time on the size of the string / set of strings S.** (Weiner, 1973) (McCreight, 1976) (Ukkonen, 1995).

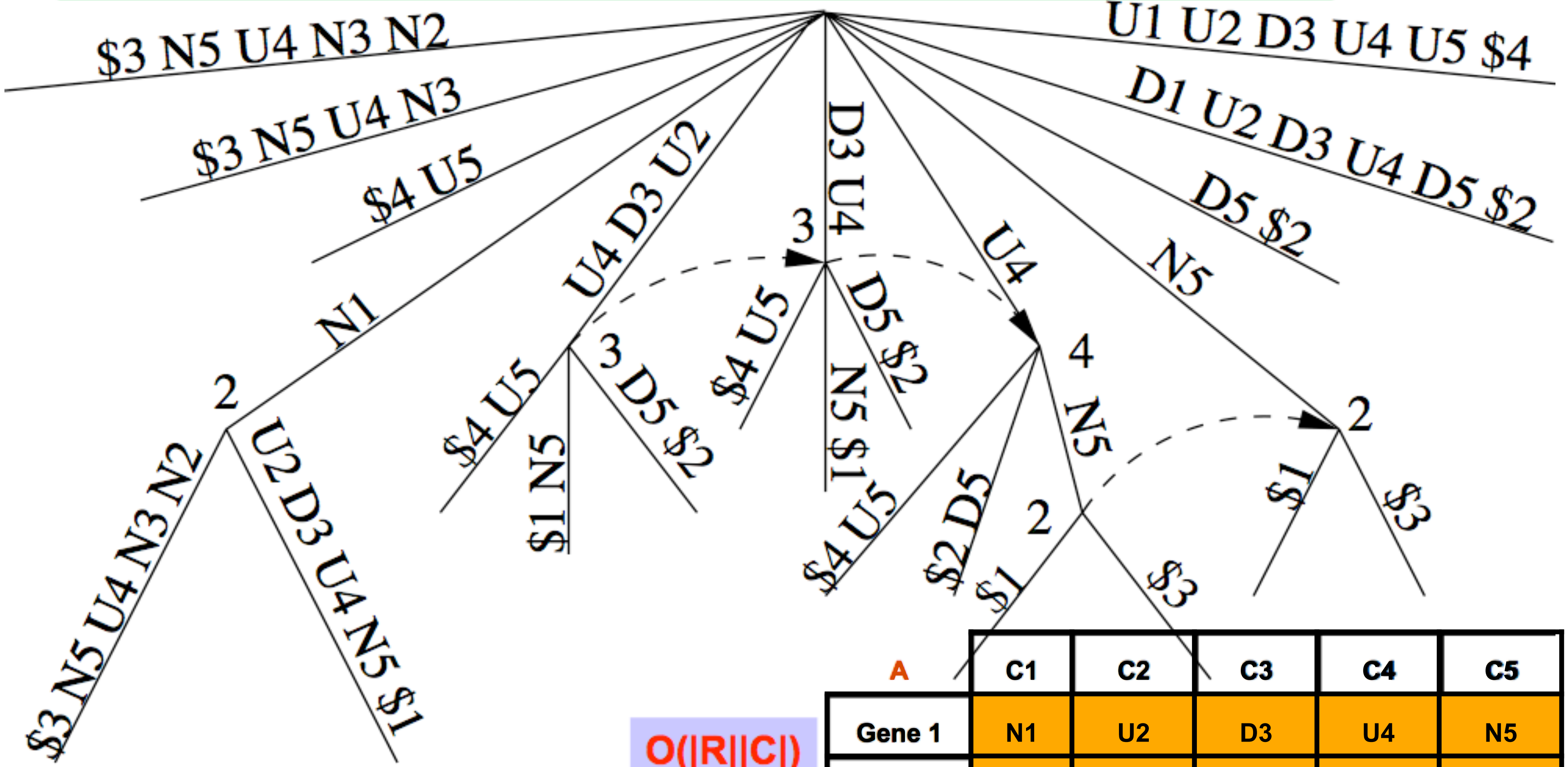
Generalized Suffix Tree with Suffix-Links

- **Ukkonen´s algorithm uses suffix-links:** Given two nodes *u* and *v*, there is a suffix link from node *v* to node *u* if the path-label of *u* represents a suffix of the path-label of *v* and the length of the path-label of *u* is exactly equal to the length of the path-label of *v* minus 1.



Generalized Suffix Tree for S1 = TACTAG and S2 = CACT

CCC-Biclustering and Suffix Trees

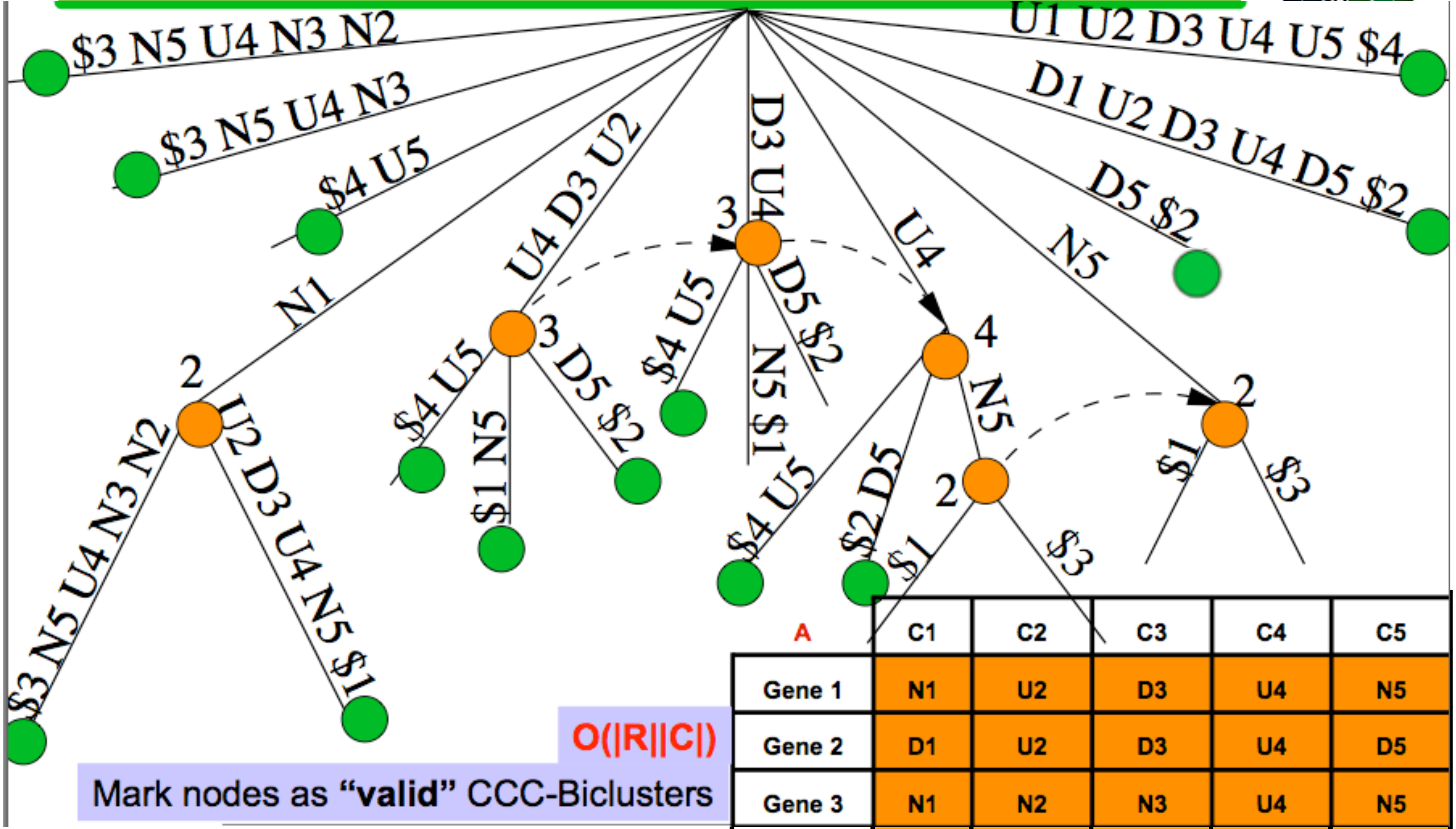


A	C1	C2	C3	C4	C5
Gene 1	N1	U2	D3	U4	N5
Gene 2	D1	U2	D3	U4	D5
Gene 3	N1	N2	N3	U4	N5
Gene 4	U1	U2	D3	U4	U5

O(|R||C|)

Build suffix tree with suffix links [Ukkonen, 1995]
Compute number of leaves of each node

CCC-Biclusters in the Suffix Tree

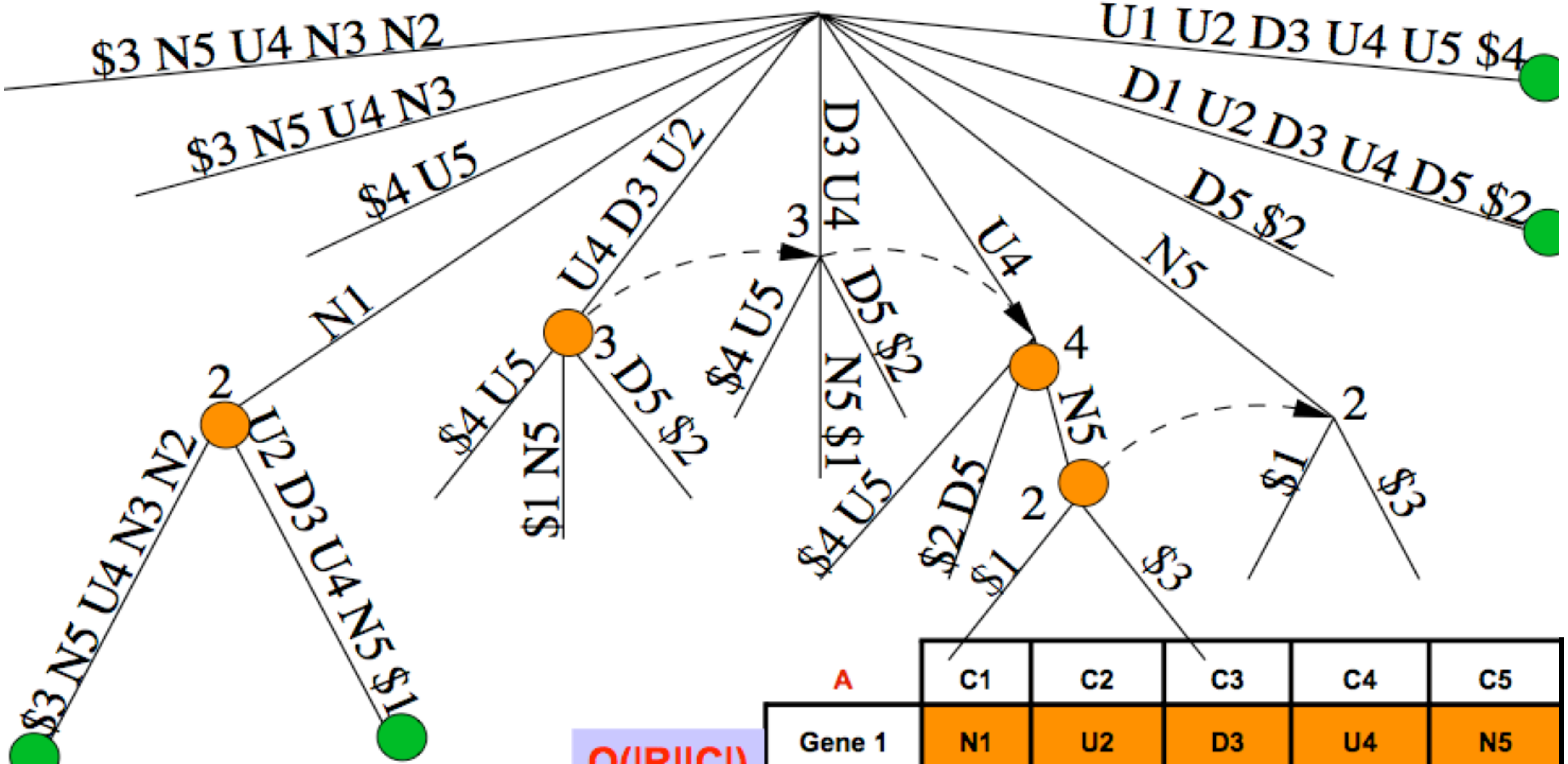


A	C1	C2	C3	C4	C5
Gene 1	N1	U2	D3	U4	N5
Gene 2	D1	U2	D3	U4	D5
Gene 3	N1	N2	N3	U4	N5
Gene 4	U1	U2	D3	U4	U5

O(|R||C|)

Mark nodes as “**valid**” CCC-Biclusters

Maximal CCC-Biclusters in the Suffix Tree

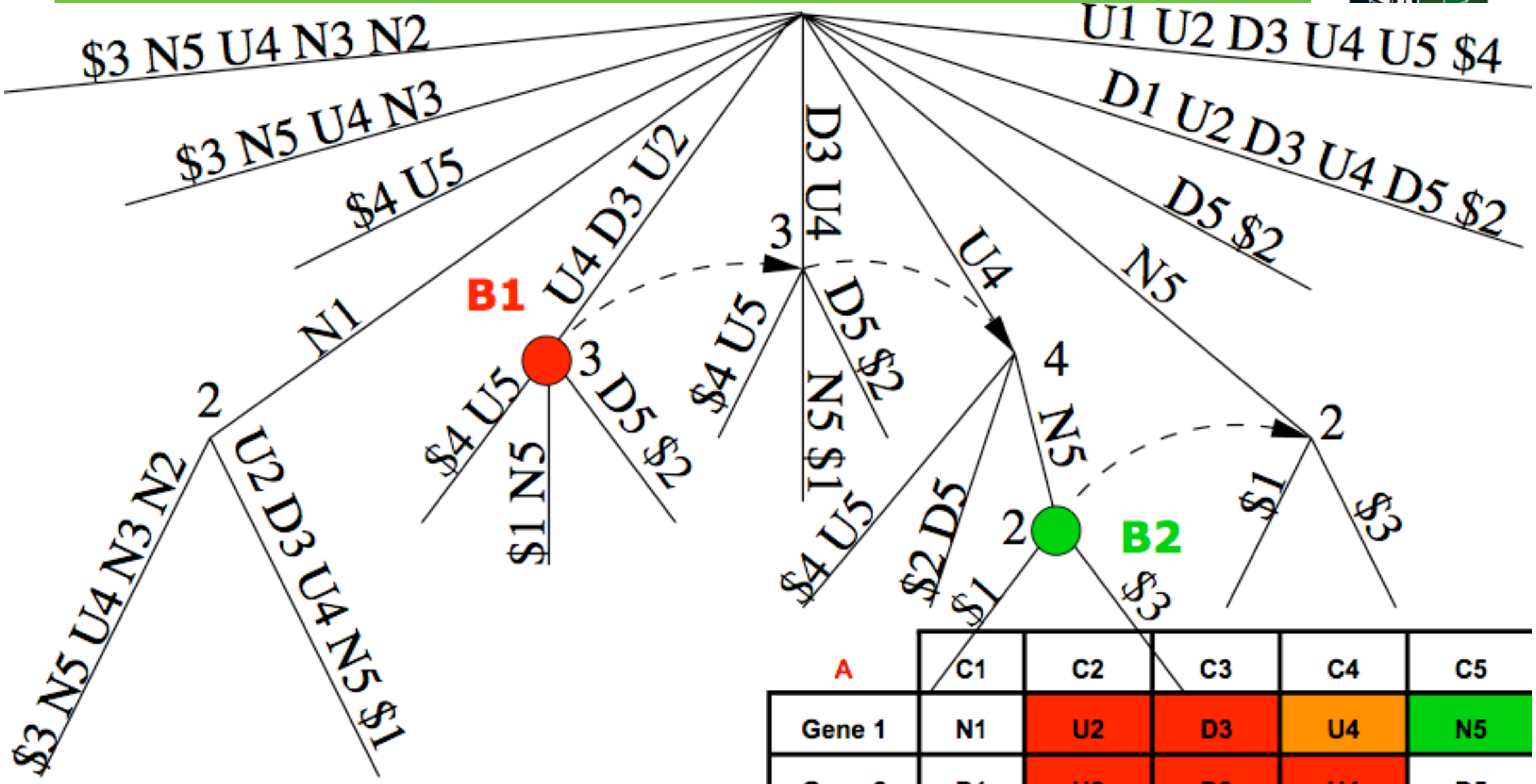


A	C1	C2	C3	C4	C5
Gene 1	N1	U2	D3	U4	N5
Gene 2	D1	U2	D3	U4	D5
Gene 3	N1	N2	N3	U4	N5
Gene 4	U1	U2	D3	U4	U5

O(|R||C|)

Mark nodes as “**invalid**” CCC-Biclusters
Report maximal CCC-Biclusters (“Valid”)

Maximal Non-Trivial CCC-Biclusters in the Suffix Tree



A	C1	C2	C3	C4	C5
Gene 1	N1	U2	D3	U4	N5
Gene 2	D1	U2	D3	U4	D5
Gene 3	N1	N2	N3	U4	N5
Gene 4	U1	U2	D3	U4	U5

Biclustering in Babelomics

- **Now:** Efficient biclustering algorithm for times series expression data - *CCC-Biclustering (Madeira et al., 2010)* extended to deal with missing values and discover opposite expression patterns (sign-changes)
- **Soon:** Efficient biclustering algorithm for non-serial expression data
- **Website:** http://beta.babelomics.bioinfo.cipf.es/
- **Tutorial:** *http://bioinfo.cipf.es/babelomicstutorial/biclustering/*