Seeding Up Genomic Data Mining via Symbolic Manipulation of Boolean Functions

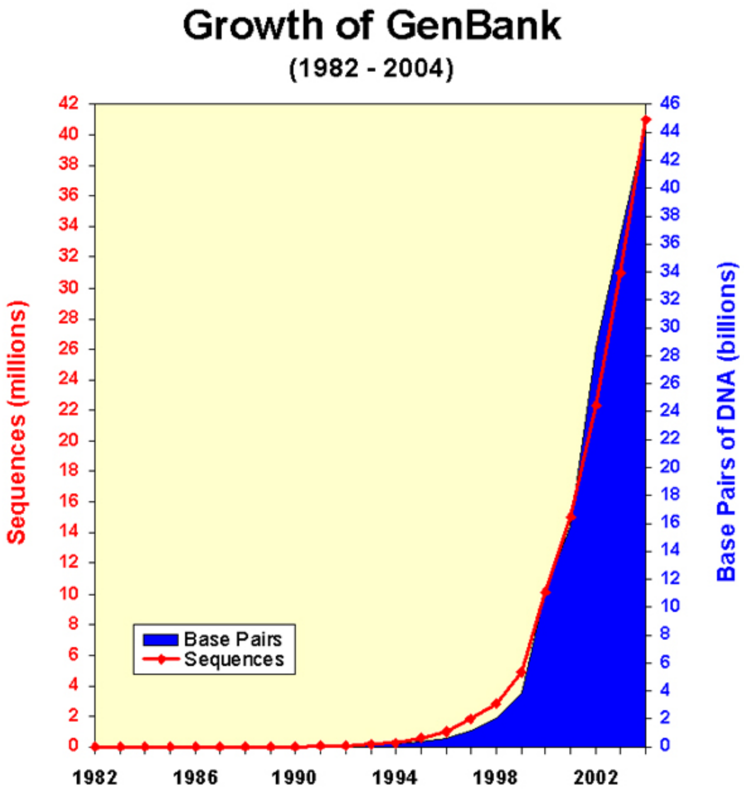
Sungroh Yoon

Computer Systems Laboratory
Stanford University

Outline

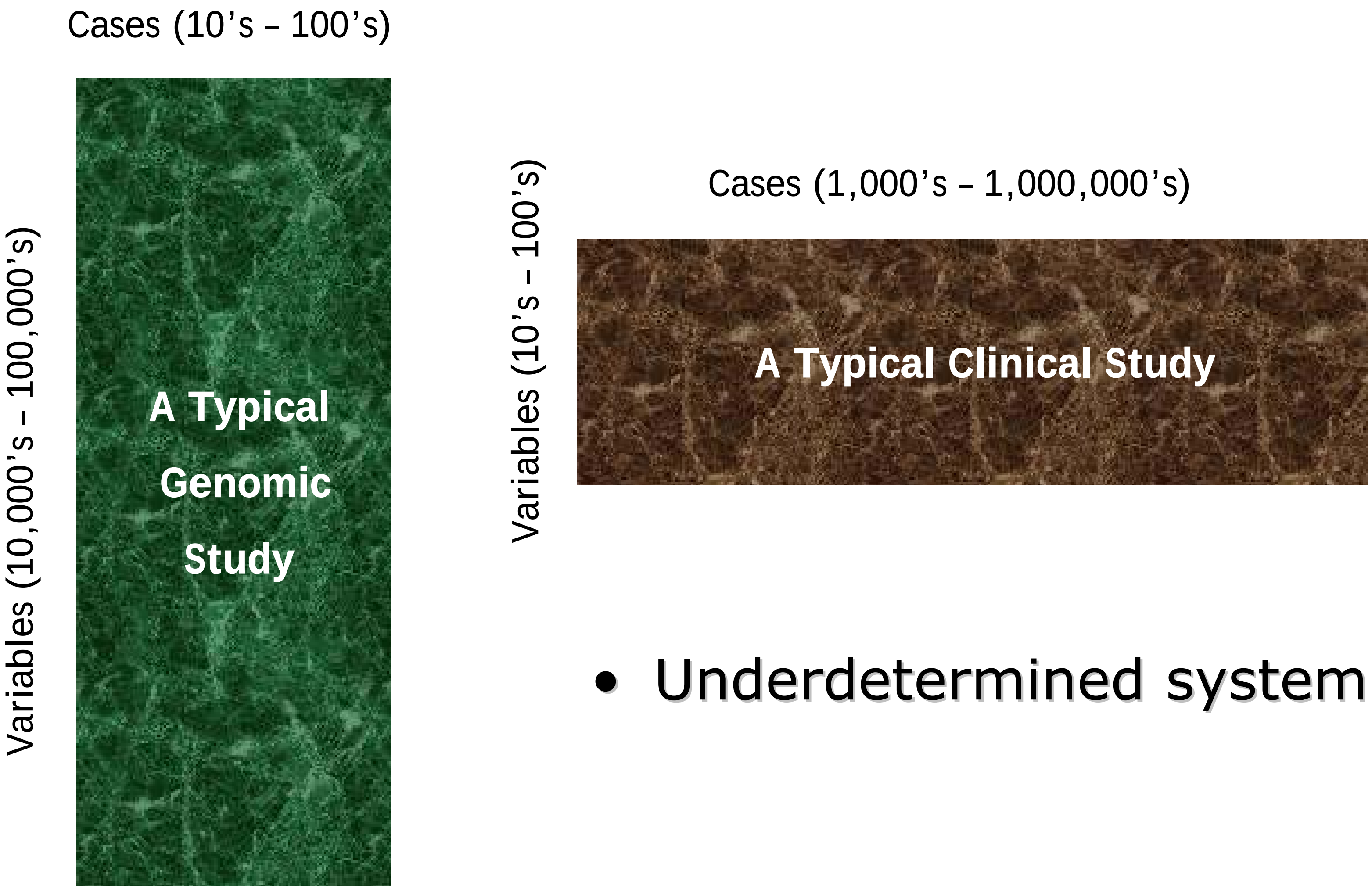
- Introduction
- Problem formulation
- Algorithm description
- Applications

Explosion of Genomic Data



- http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html

Challenges in Genomic Data Analysis



Cases (10’s – 100’s)

Variables (10,000’s – 100,000’s)

A Typical Genomic Study

Cases (1,000’s – 1,000,000’s)

Variables (10’s – 100’s)

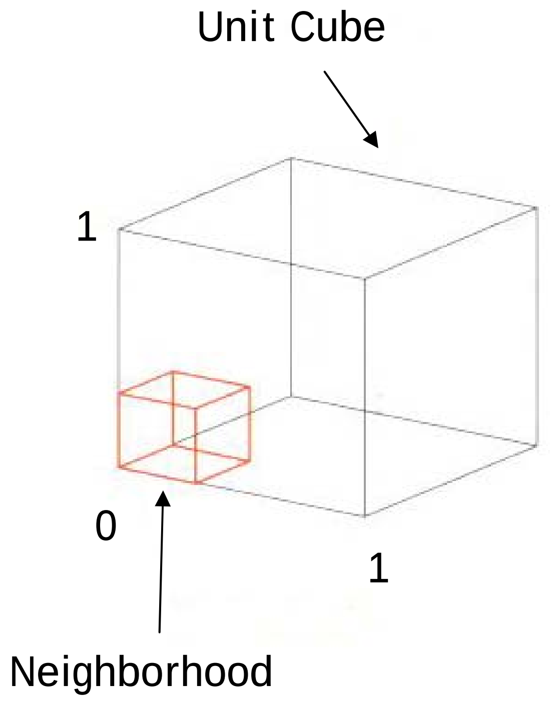
A Typical Clinical Study

- Underdetermined system

Kohane et al., *Microarrays for Integrative Genomics*, 2003

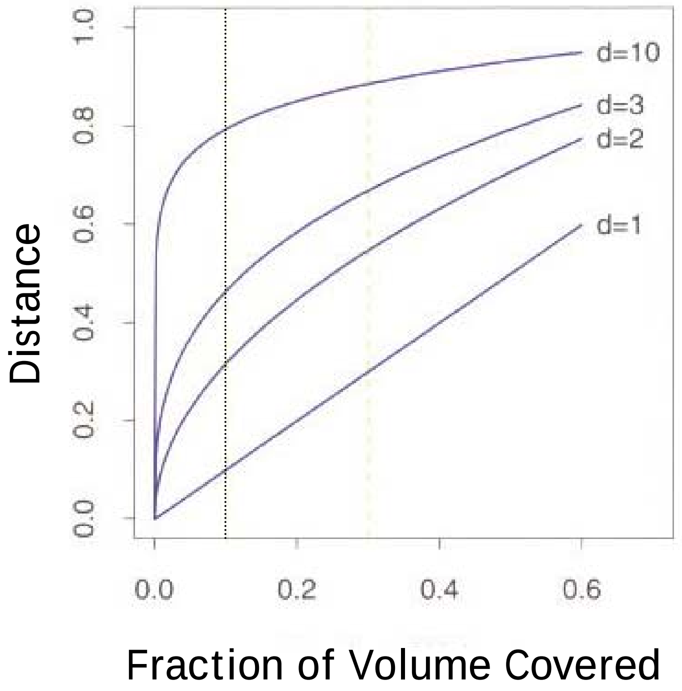
Challenges in Genomic Data Analysis

- The curse of dimensionality
- Example: k-NN



Unit Cube

Neighborhood



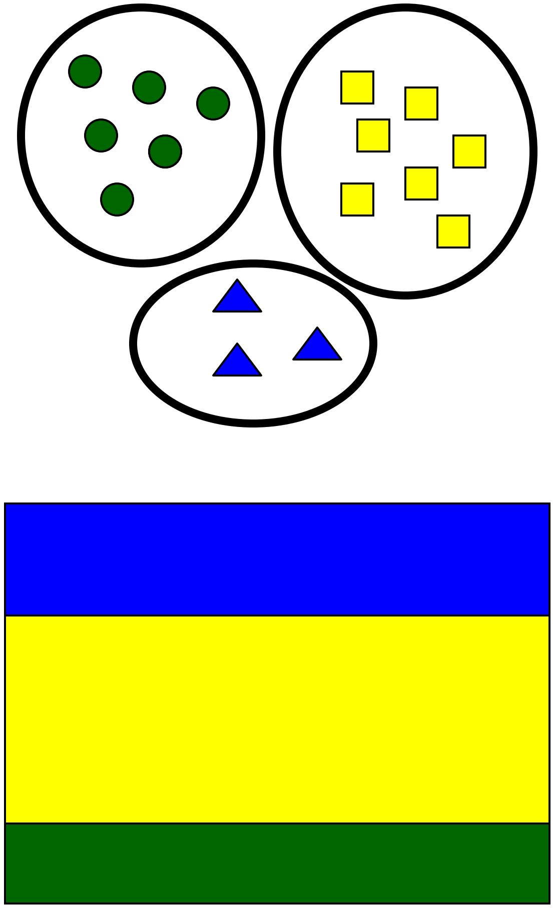
Hastie et al., *The Elements of Statistical Learning*, 2001

Biclustering

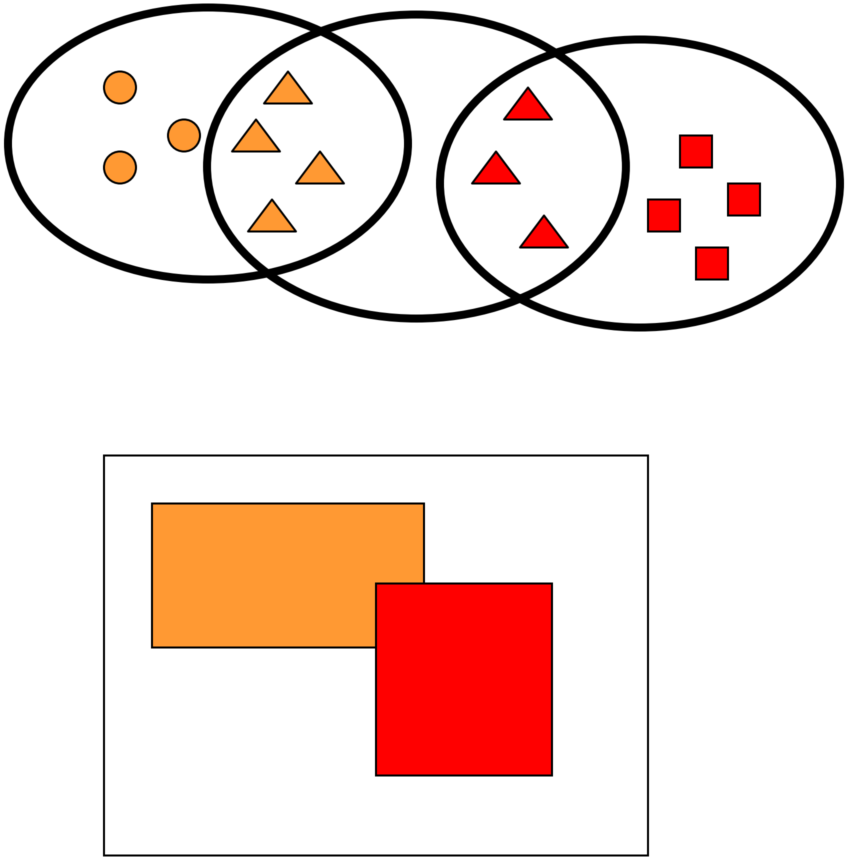
- Idea
  - Focus on subsets of data
  - Simultaneous clustering of objects and features
- Subspace clustering + 2-way clustering
- Allow overlaps

Clustering vs. biclustering

- Clustering



- Biclustering



Biclustering

- Pros
  - Can handle the curse better
    - Some desirable statistical properties reported
  - Finer, more localized analysis
    - Samples/experimental conditions can be diverse
  - Overlaps allowed
    - Genes have multiple functions
- Cons
  - Reliable statistics needed for validation
  - Inherently intractable problem

Main Contributions

- Mathematical characterization of the biclustering problem
- An exact and scalable algorithm
  - Leveraged by symbolic manipulation of Boolean functions
- Applications
  - DNA microarray data analysis
  - Predicting gene regulatory modules
  - Correlating clinical parameters with genes

Outline

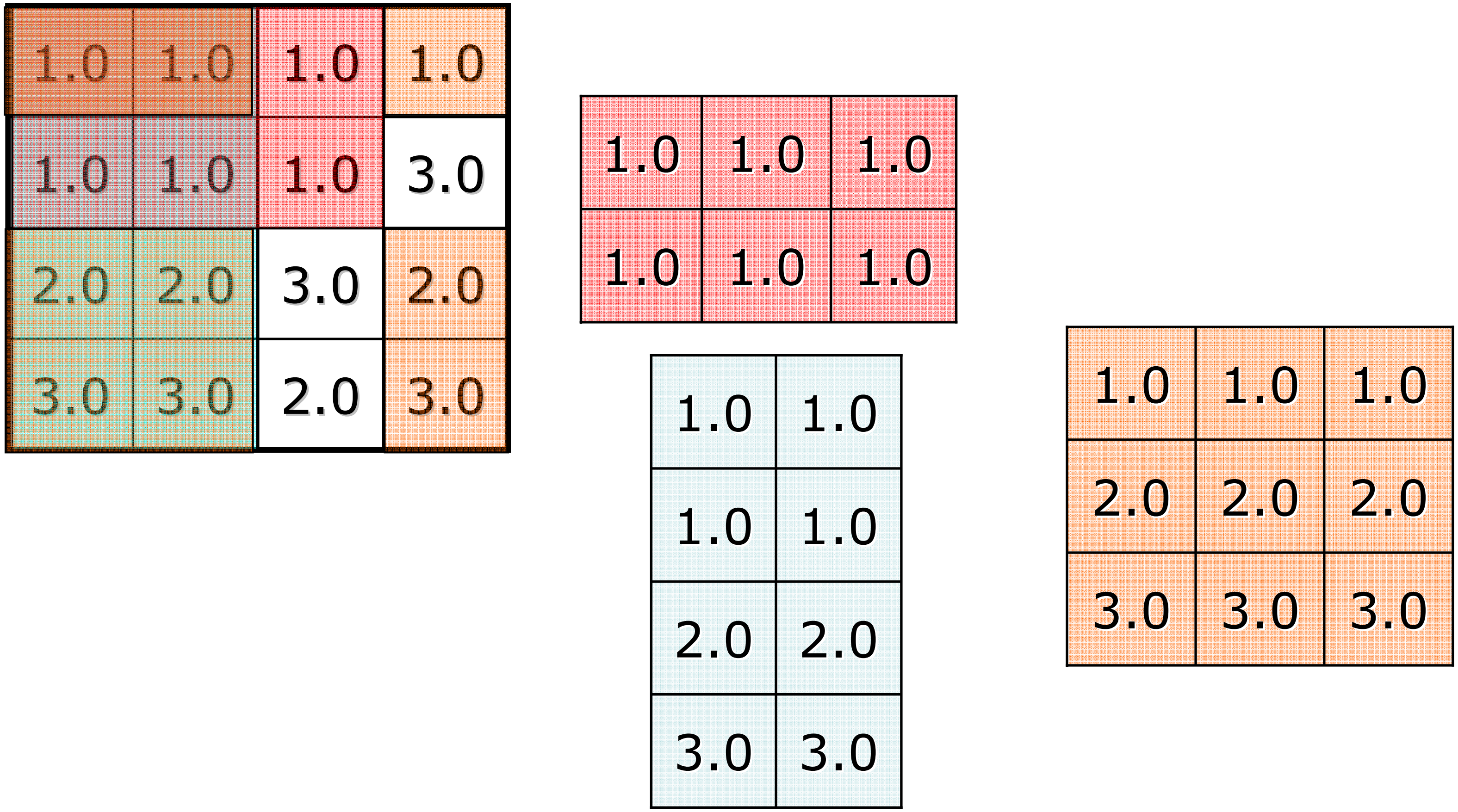
- Introduction
- Problem formulation
  - – Homogeneous biclusters
  - – 3 types of homogeneous biclusters
- Algorithm description
- Applications

Biclustering Problem

- Input
  - A=(R,C), a matrix of real numbers
  - D, a specific condition giving some semantics
- Output
  - Biclusters B=(I,J)
    - Submatrices of A
    - Satisfy the condition D
    - Can overlap with each other

Example

- D: the values on each row are constant



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

1.0	1.0	1.0
1.0	1.0	1.0

1.0	1.0
1.0	1.0
2.0	2.0
3.0	3.0

1.0	1.0	1.0
2.0	2.0	2.0
3.0	3.0	3.0

Homogeneous Biclusters

- Given matrix A and condition D
- Suppose B is a bicluster in A under D
- B is called *homogeneous* if
  - Any sub-bicluster of B also satisfies D
- Example
  - D: the values on each row are constant

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

→

1.0	1.0	1.0
2.0	2.0	2.0
3.0	3.0	3.0

Homogeneous Biclusters

- Desirable properties
  - More intuitive
  - Allow us to devise a more efficient algorithm
    - Optimal substructure for dynamic programming
  - Can be statistically better
    - Smaller residues (in ANOVA theory)
- Many examples in the literature
  - xMOTIF, δ-valid kj pattern, GEMS
  - OPSM, OP-cluster
  - δ-pCluster

Our Method

- Applicable to finding homogeneous biclusters of any definition
- Three specific examples
  - Type 1
    - Biclusters with constant values on rows
  - Type 2
    - Biclusters in which the order of values on each row is preserved
  - Type 3
    - Biclusters with coherent values

Type 1

- Definition: RANGE
  - RANGE(<1,2,3,4,5>) = 4
- Type 1 homogeneous bicluster
  - A submatrix (I,J)
  - Given τ >= 0, $\forall i \in I, \text{RANGE}(\{a_{ij} | \forall j \in J\}) \leq \tau.$
- Related examples
  - xMOTIF, δ-valid kj pattern, GEMS

Type 1 Bicluster

$\tau = 0.5$

1.4	1.0	1.2	1.7
3.0	3.1	3.3	3.0
2.0	1.9	2.4	2.5
3.2	3.7	2.0	3.0

$1.4 - 1.0 = 0.4 <= 0.5$

$3.3 - 3.0 = 0.3 <= 0.5$

$2.4 - 1.9 = 0.5 <= 0.5$

Type 2

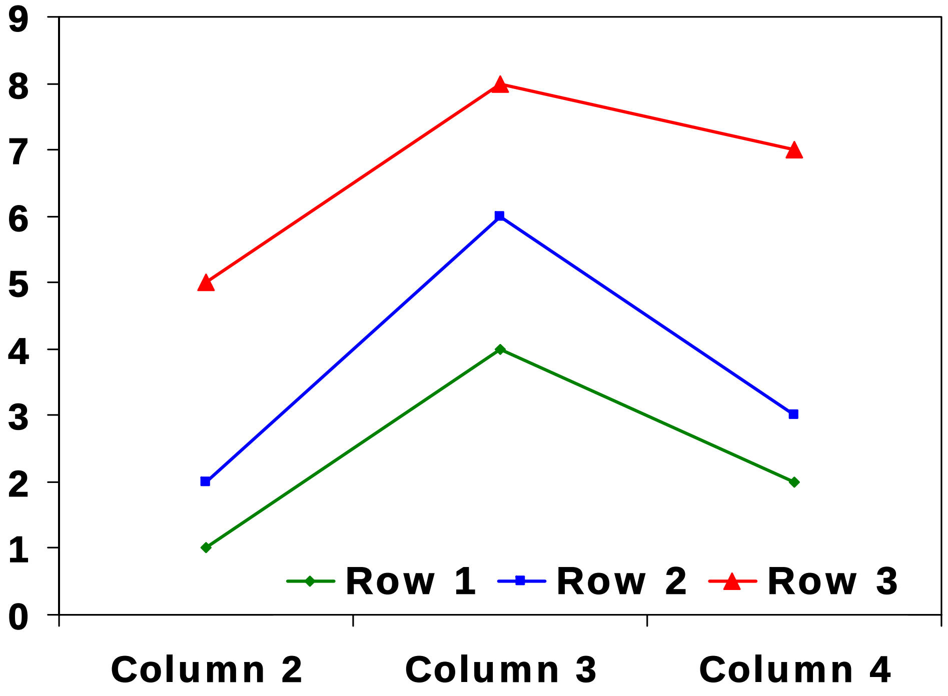
- Given a data matrix A = (R,C)
  - Let J ✪ C be a set of size s >= 2
  - Let π = (p_1,p_2,…,p_s) be a linear ordering of J
- Type 2 homogeneous bicluster
  - A submatrix (I,J)
  - The order of the values on rows are preserved
  - That is, $i \in I,\ a_{ip_1} > a_{ip_2} > \cdots > a_{ip_s}.$
- Related examples
  - OPSM, OP-cluster

Type 2 Bicluster

- R={1,2,3}, C={1,2,3,4}

	1	2	3	4
1	3.0	1.0	4.0	2.0
2	7.0	2.0	6.0	3.0
3	6.0	5.0	8.0	7.0

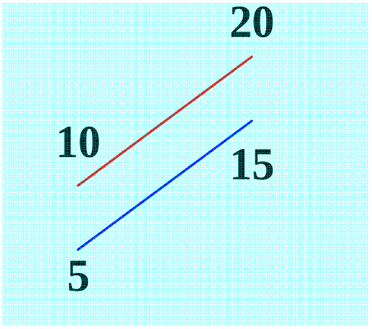
- $i \in \{1, 2, 3\}$,
  $a_{i3} > a_{i4} > a_{i2}$.



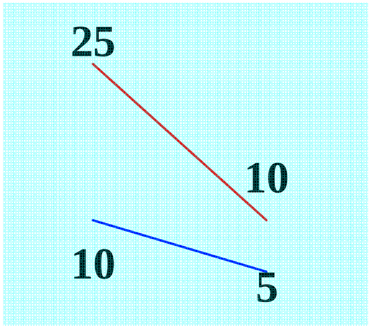
pScore (Wang et al, 2002)

- Measure of coherence
  - Defined over 2 ◉ 2 matrices

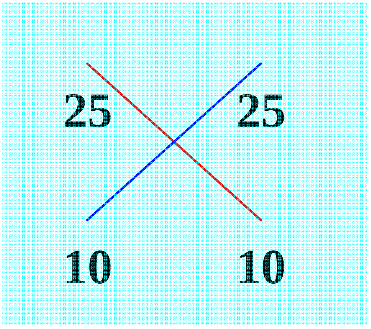
$$
\begin{aligned}
pScore\left(\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix}\right) &= |(x_1 - y_1) - (x_2 - y_2)| \\
&= |(x_1 - x_2) - (y_1 - y_2)|
\end{aligned}
$$



pScore = 0



pScore = 10



pScore = 30

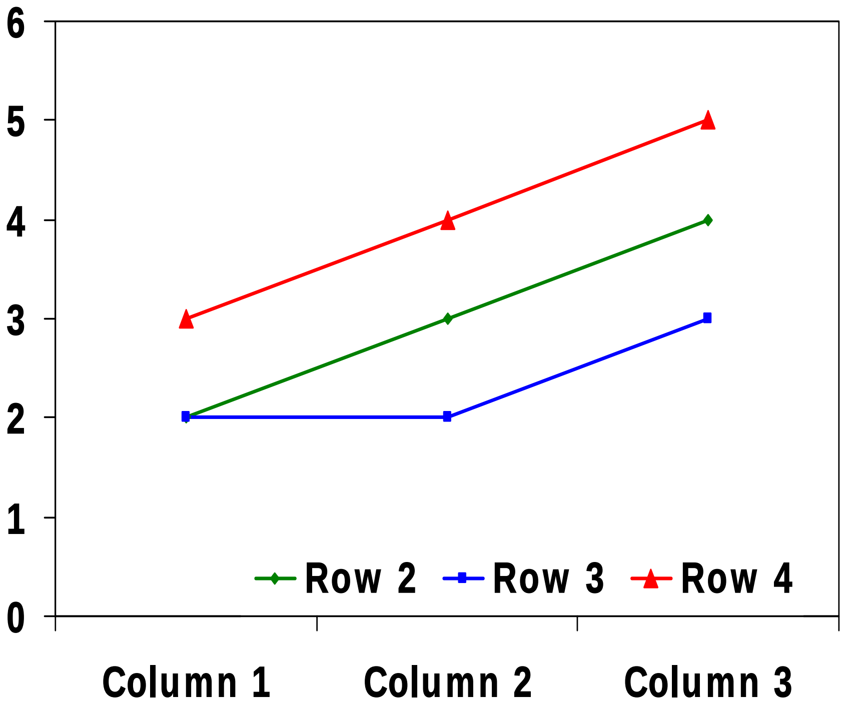
Type 3

- Type 3 homogeneous bicluster
  - A submatrix (I,J)
  - Given $\tau >= 0$
    - Every 2 x 2 submatrix X has $\mathrm{pScore}(X) \leq \tau$
- Related examples
  - δ-bicluster, FLOC cluster, δ-pCluster

Type 3 Bicluster

- $\tau = 1$

6.0	9.0	2.0	5.0
2.0	3.0	4.0	3.0
2.0	2.0	3.0	6.0
3.0	4.0	5.0	2.0



$$pScore\left(\begin{bmatrix} 2 & 3 \\ 2 & 2 \end{bmatrix}\right) = |(2-2)-(3-2)| = 1 \leq \tau$$

Problem Statement

- Given an input data matrix $A = (R, C)$ and type $t \in \{Type1, Type2, Type3\}$

- The problem of biclustering is to find all maximal biclusters of type t appearing in A
  - A bicluster is *maximal* if it is not contained by other biclusters

Outline

- Introduction
- Problem formulation
- Algorithm description
  - Overview
  - Step 1: finding seeds
  - Step 2: deriving biclusters from seeds
- Applications

Overview

Two-step process

STEP 1

- User-specified front-end
- AD Converter
- Generate “seeds”
- Determines bicluster type

Type 1

Type 2

Type 3

xMOTIF

δ valid kj pattern

GEMS

OPSM

OP-cluster

: :

STEP 2

- Powerful biclustering engine
- DSP
- Derive biclusters from seeds
- Common for all bicluster type

Step 1: Seeds

- Biclusters of the minimum number of rows

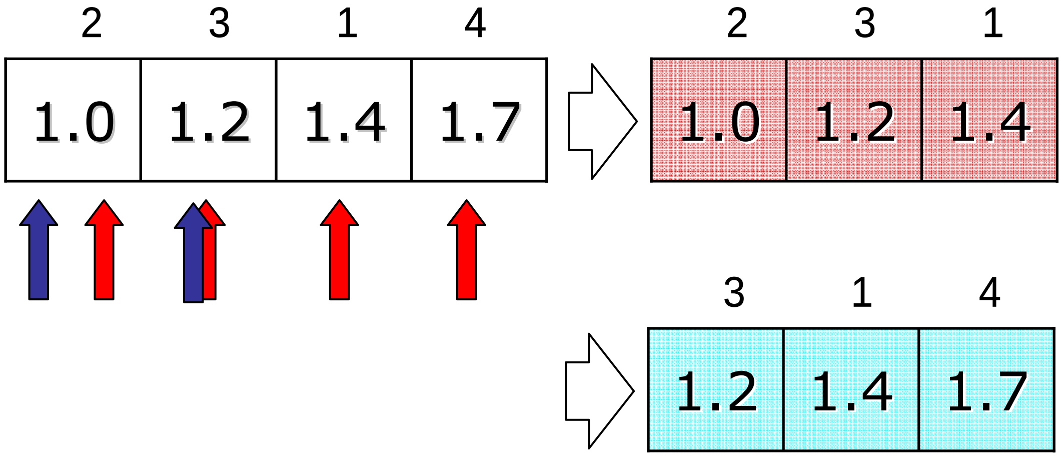
Type 1	(1 x k) matrix
Type 2	(2 x k) matrix
Type 3	(2 x k) matrix

- Finds only maximal seeds
- Polynomial-time algorithm

Finding Type 1 Seeds

- Type 1 seed: a row vector with near constant values
- Example (τ = 0.5)

	1	2	3	4
1	1.4	1.0	1.2	1.7
2	3.0	3.1	3.3	3.0
3	2.0	1.9	2.4	2.5
4	3.2	3.7	2.0	3.0



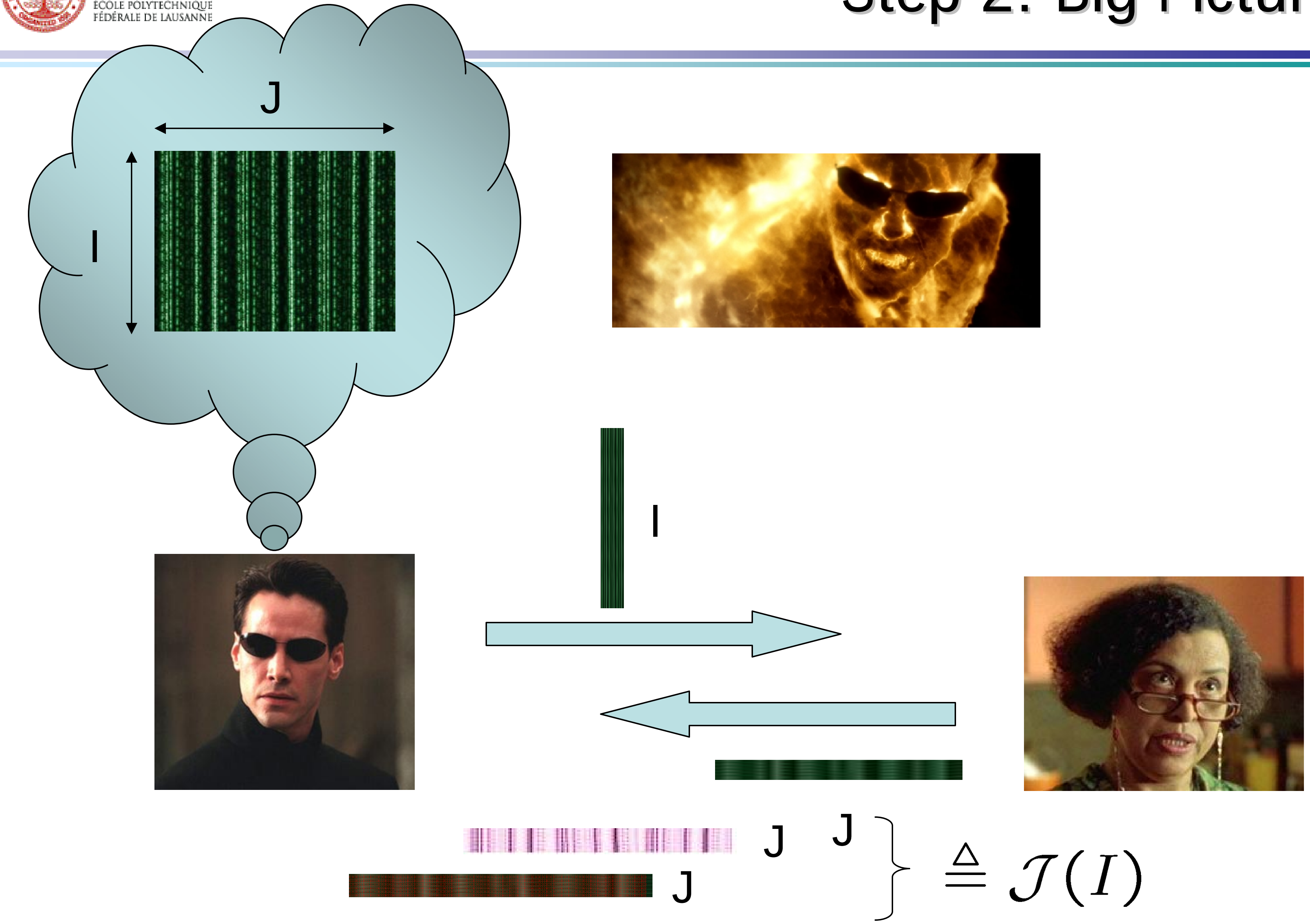
2	3	1	4
1.0	1.2	1.4	1.7

2	3	1
1.0	1.2	1.4

3	1	4
1.2	1.4	1.7

- Two seeds for row 1
  - ({1},{1,2,3})
  - ({1},{1,3,4})

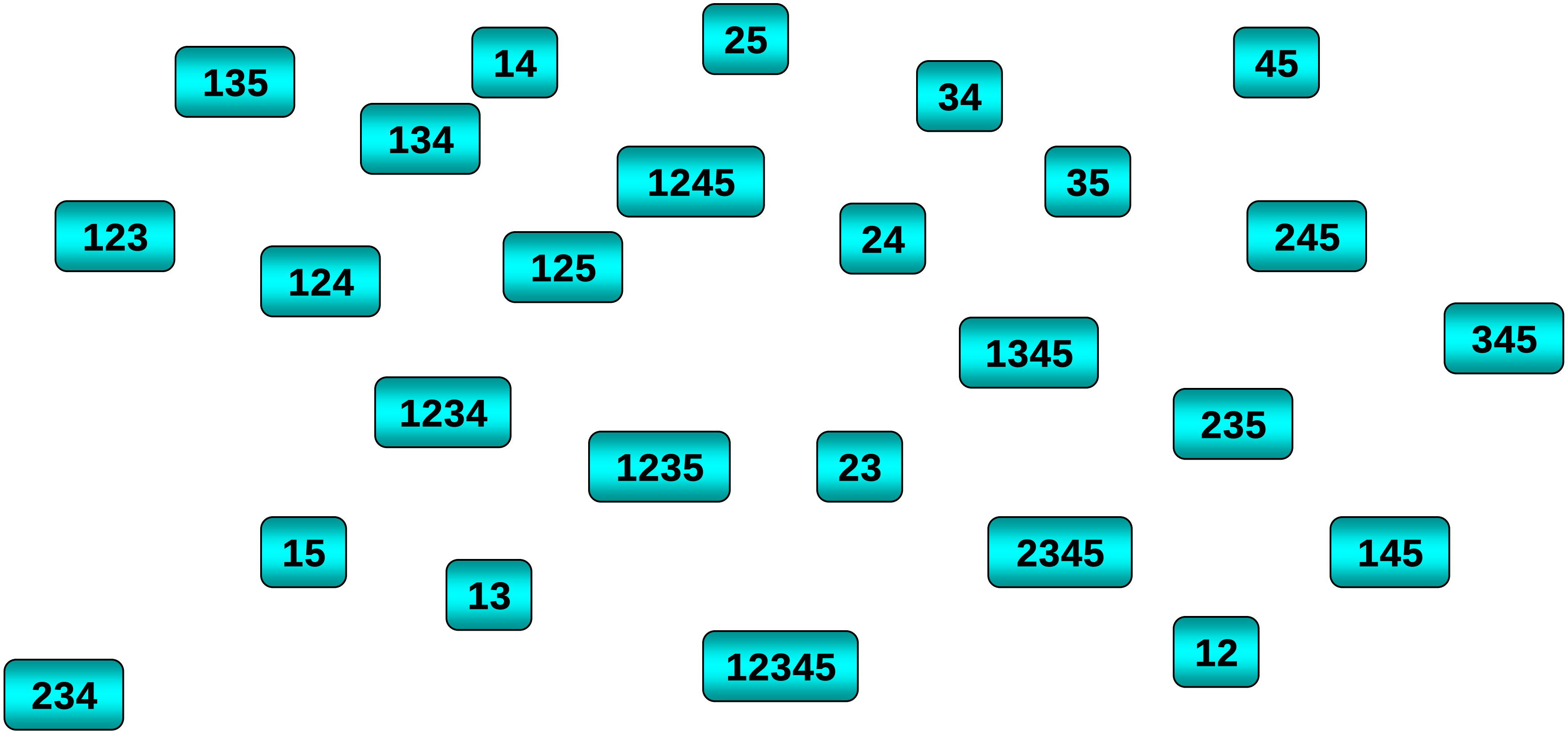
Step 2: Big Picture



$$\triangleq \mathcal{J}(I)$$

Determining I

$$I \in 2^R,\ R = \{1,2,3,4,5\}$$



25

14

45

135

34

134

1245

35

123

245

24

125

124

345

1345

1234

235

1235

23

15

2345

145

13

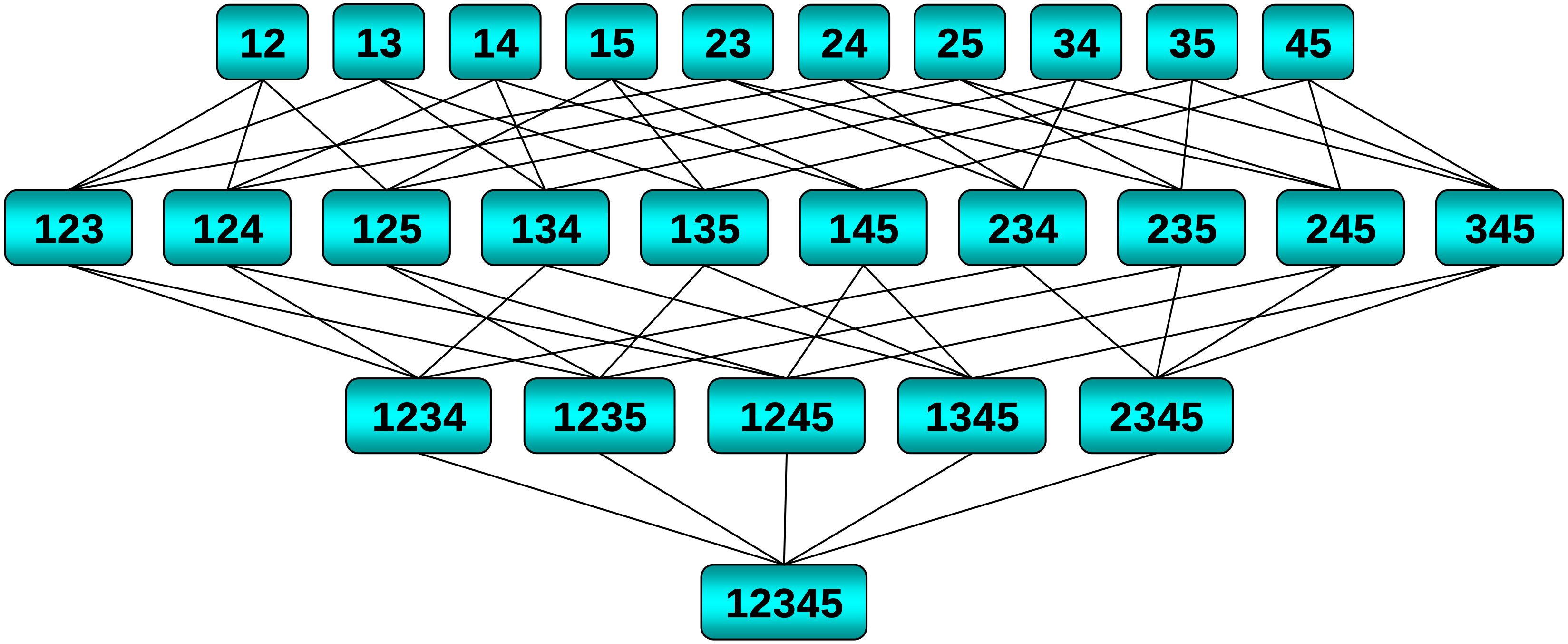
12

12345

234

Determining I

- Lattice for $I \in 2^R,\ R = \{1,2,3,4,5\}$



12 13 14 15 23 24 25 34 35 45

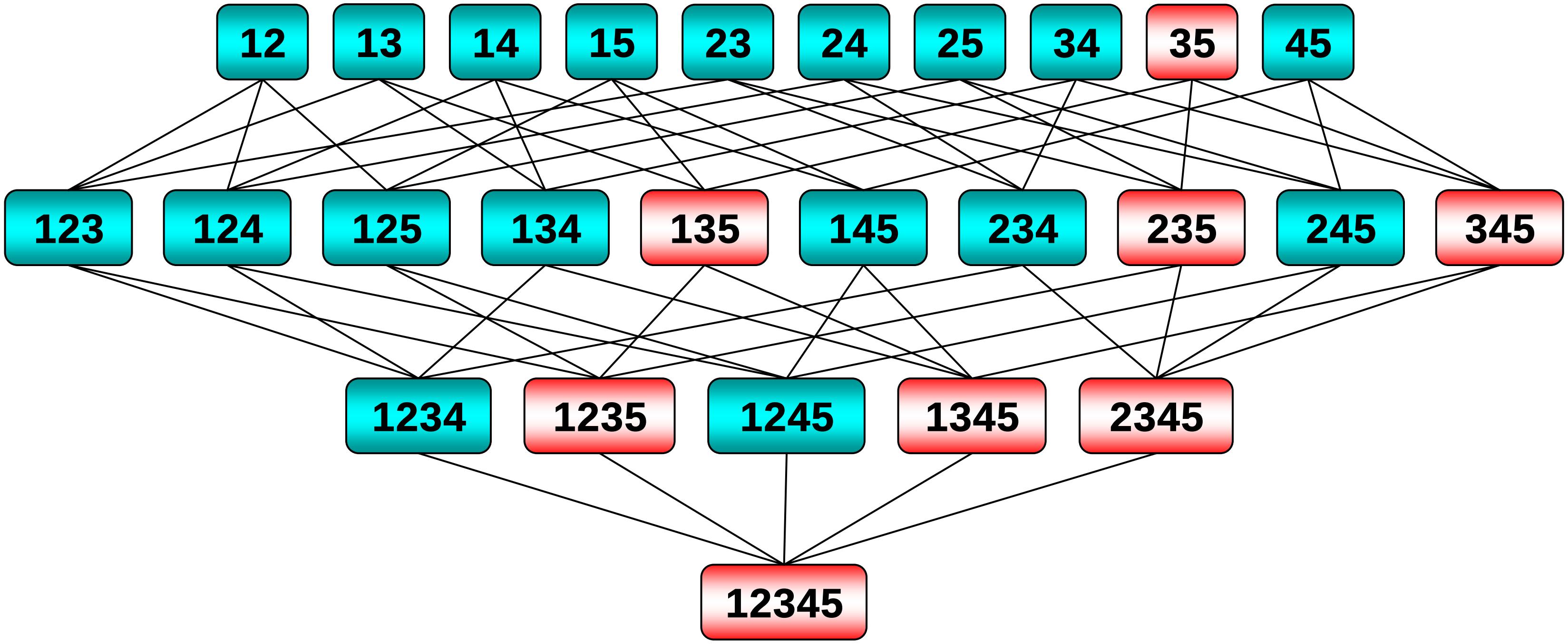
123 124 125 134 135 145 234 235 245 345

1234 1235 1245 1345 2345

12345

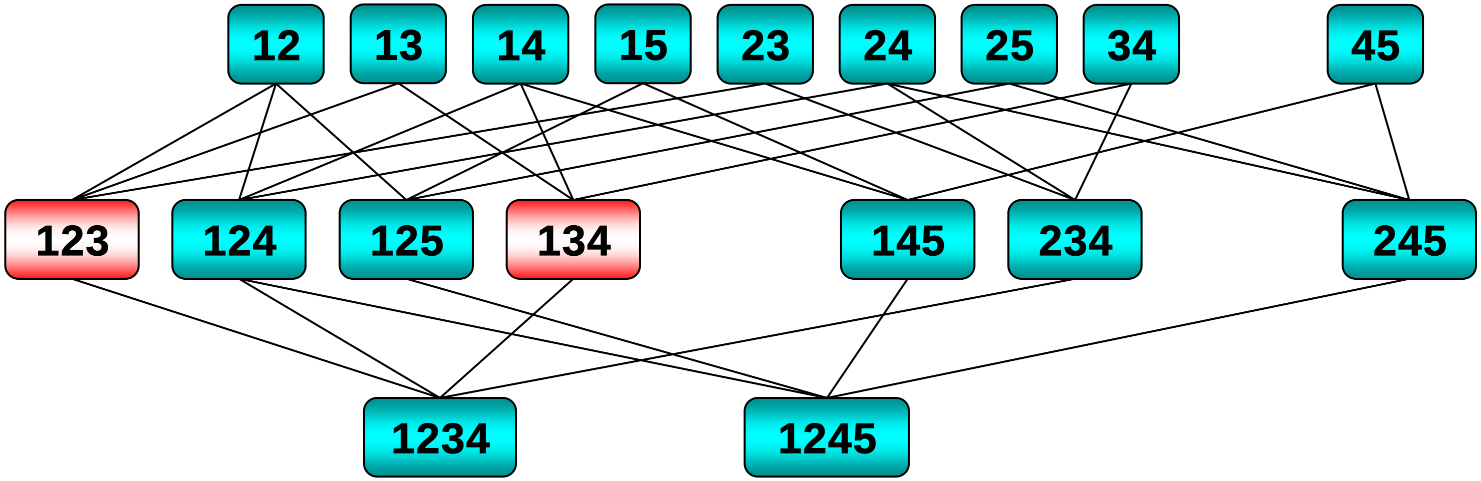
Determining I

- Suppose that $\mathcal{J}(\{3,5\}) = \emptyset$



Determining I

- Also, $\mathcal{I}(\{1,2,3\}) = \mathcal{I}(\{1,3,4\}) = \emptyset$



12 13 14 15 23 24 25 34 45

123 124 125 134 145 234 245

1234 1245

Most Cited CS Papers

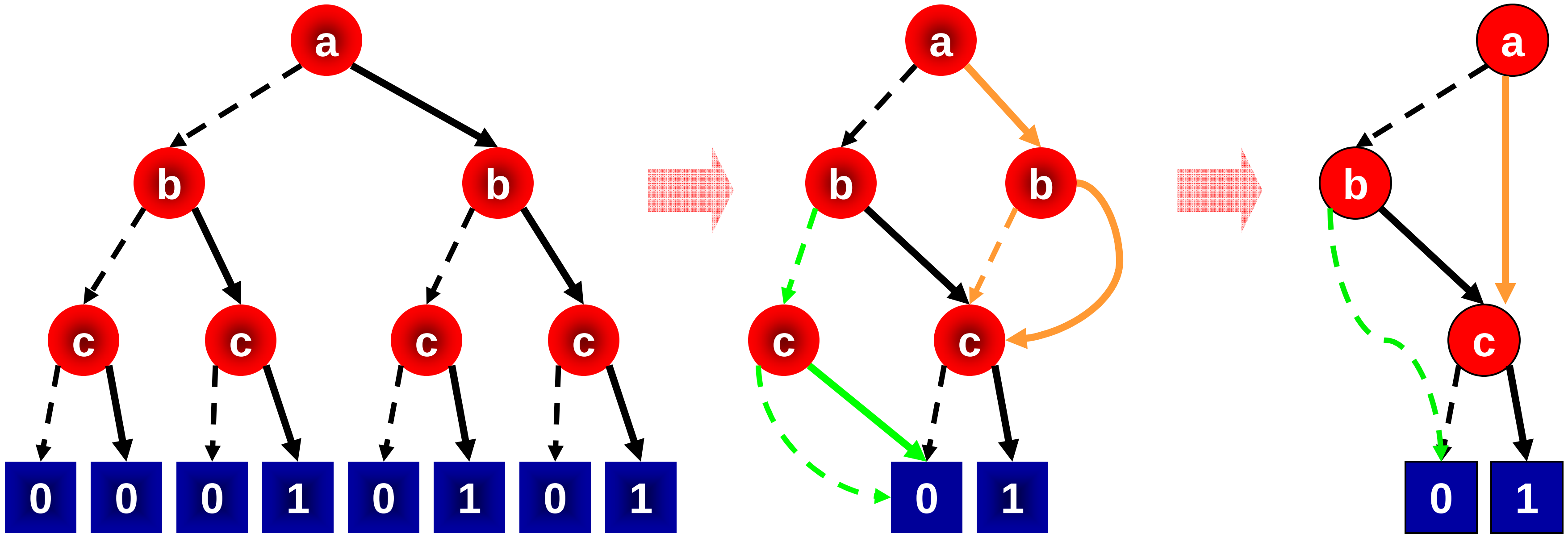
- http://citeseer.ist.psu.edu/source.html (as of July 2004)

1. *Graph-Based Algorithms for Boolean Function Manipulation* - Bryant (1986)
2. *Optimization by Simulated Annealing* - Kirkpatrick, Gelatt, Jr., Vecchi (1983)
3. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems* - Rivest, Shamir, Adleman (1978)

14. *Fast Algorithms for Mining Association Rules* - Agrawal, Srikant (1994)
15. *The Java Language Specification* - Gosling, Joy, Steele (1996)

Reduced Ordered BDDs (Bryant)

- Ordered
- Reduced
  - 1. Merge equivalent sub-trees
  - 2. Remove nodes with identical children



Zero-suppressed BDDs (Minato)

- Optimized for sets of combinations

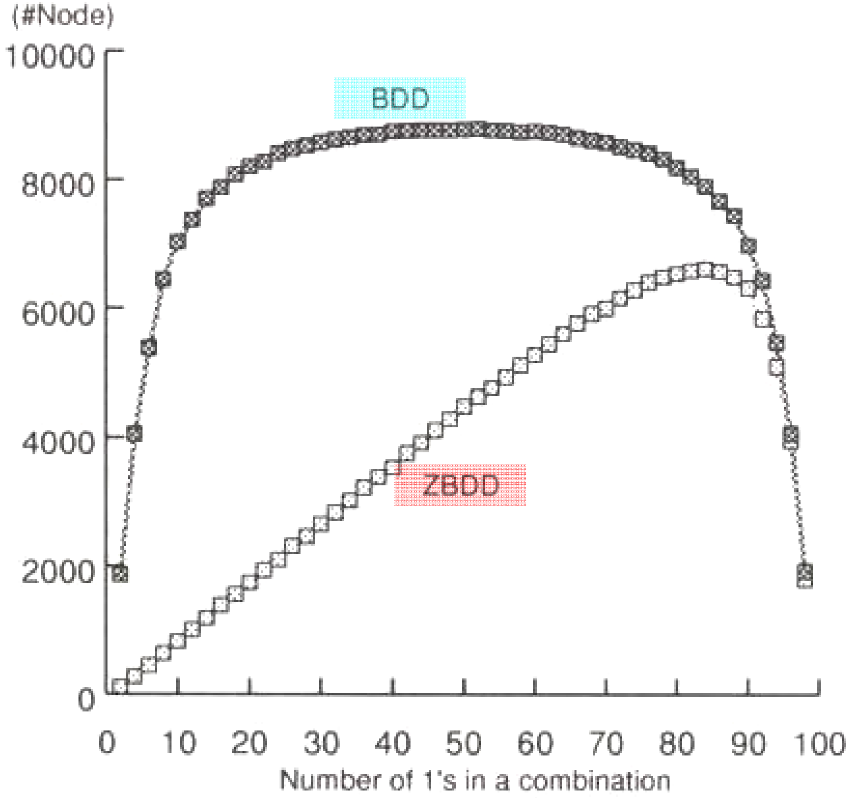


Figure 6.6 Comparison of BDDs and ZBDDs.

S. Minato, *BDDs and Applications for VLSI CAD*, Kluwer, 1996

Definition

- Given $A = (R, C)$, an input data matrix, and $\mathfrak{D}$, a specific definition of a bicluster, $\mathfrak{R}_{\mathfrak{D}}$ is a binary relation on $2^R \times 2^C$:

  $$\mathfrak{R}_{\mathfrak{D}} = \{(I, J) | \text{The pair } (I, J) \text{ is a bicluster in } A \text{ under } \mathfrak{D}\}.$$

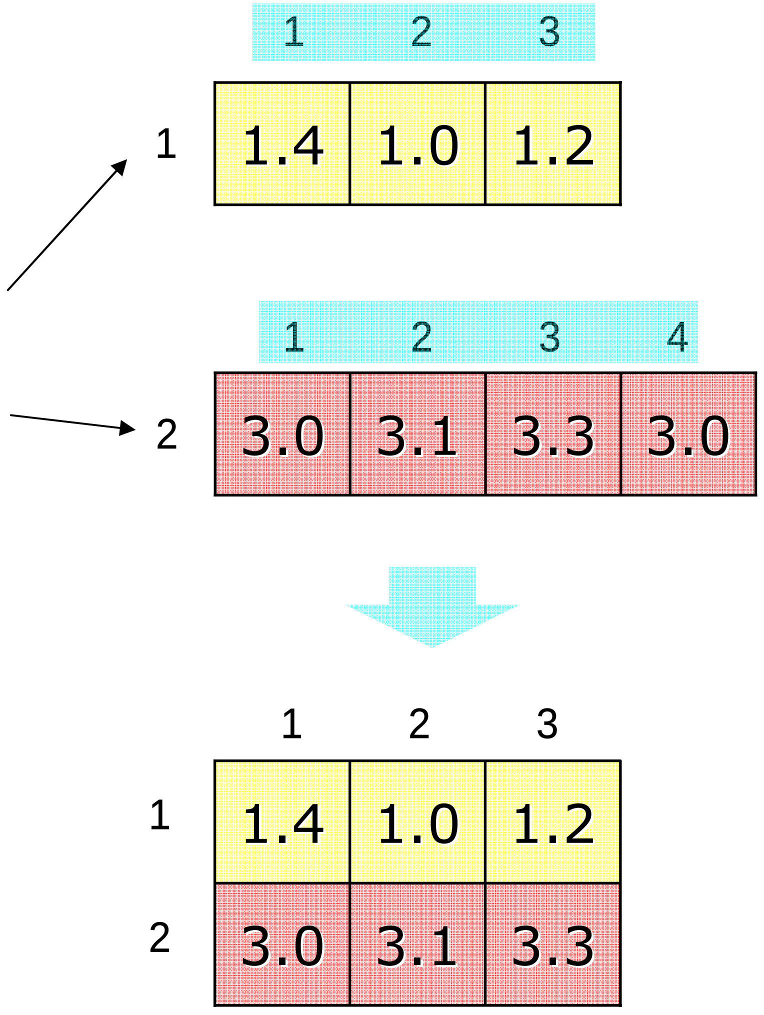
- Given matrix $A = (R, C)$, $\mathcal{J}$ is a function that maps $I \in 2^R$ to the image $\mathcal{J}(I)$, where

  $$\mathcal{J}(I) = \{J \in 2^C | (I, J) \in \mathfrak{R}_{\mathfrak{D}} \text{ and } \nexists J' \supset J \text{ s.t. } (I, J') \in \mathfrak{R}_{\mathfrak{D}}\}.$$

Example

- $\tau=0.5$, $t=1$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1.4 & 1.0 & 1.2 & 1.7 \\ 2 & 3.0 & 3.1 & 3.3 & 3.0 \\ 3 & 2.0 & 1.9 & 2.4 & 2.5 \\ 4 & 3.2 & 3.7 & 2.0 & 3.0 \end{array}$$



$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1.4 & 1.0 & 1.2 \end{array} \qquad J(\{1\})$$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 2 & 3.0 & 3.1 & 3.3 & 3.0 \end{array} \qquad J(\{2\})$$

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1.4 & 1.0 & 1.2 \\ 2 & 3.0 & 3.1 & 3.3 \end{array}$$

- I = {1,2}
- J = ?

$$J(\{1,2\}) = J(\{1\}) \cap J(\{2\})$$

Example

- τ=1, t=3

	1	2	3	4
1	6.0	9.0	2.0	5.0
2	2.0	3.0	5.0	6.0
3	2.0	2.0	3.0	4.0
4	1.0	0.0	1.0	2.0

- I = {2,3,4}
- J = ?

$J(\{2,3\})$

	1	2	3	4
2	2.0	3.0	5.0	6.0
3	2.0	2.0	3.0	4.0

$J(\{2,4\})$

	2	3	4
2	3.0	5.0	6.0
4	0.0	1.0	2.0

$J(\{3,4\})$

	1	2	3	4
3	2.0	2.0	3.0	4.0
4	1.0	0.0	1.0	2.0

$$J(\{2,3,4\}) = J(\{2,3\}) \cap J(\{2,4\}) \cap J(\{3,4\})$$

Definition

- The *pairwise intersection* of two sets of sets $\mathcal{A}$ and $\mathcal{B}$, denoted by $\mathcal{A} \otimes \mathcal{B}$, is defined as

  $\mathcal{A}\otimes\mathcal{B} = \{I | I = A \cap B, \ \forall A \in \mathcal{A}$ and $\forall B \in \mathcal{B},$ and I is maximal$\}$.

- $\{\{0,1,2\},\{2,3,4\}\} \otimes \{\{0,2\},\{4,5\}\} = \{\{0,2\},\{4\}\}$

- The pairwise intersection of the n sets of sets $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ is denoted by

$$\mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \cdots \otimes \mathcal{A}_n = \bigotimes_{i=1}^{n} \mathcal{A}_i .$$

Theorem

- Let $\mathcal{J}_1$, $\mathcal{J}_2$, and $\mathcal{J}_3$ denote the function $\mathcal{J}$ for Types 1, 2, and 3, respectively.
- Given input data $A = (R, C)$, the image of $I \in 2^R$, or $\mathcal{J}(I)$, can be represented as follows.
  - When the set I has only one or two elements:

$$\begin{aligned}
\mathcal{J}_1(\{r\}) &= \{J|\text{The pair } (\{r\}, J) \text{ is a Type 1 seed for row } r \in R\} \\
\mathcal{J}_2(\{q,r\}) &= \{J|\text{The pair } (\{q,r\}, J) \text{ is a Type 2 seed for rows } q, r \in R\} \\
\mathcal{J}_3(\{q,r\}) &= \{J|\text{The pair } (\{q,r\}, J) \text{ is a Type 3 seed for rows } q, r \in R\}
\end{aligned}$$

  - Otherwise:

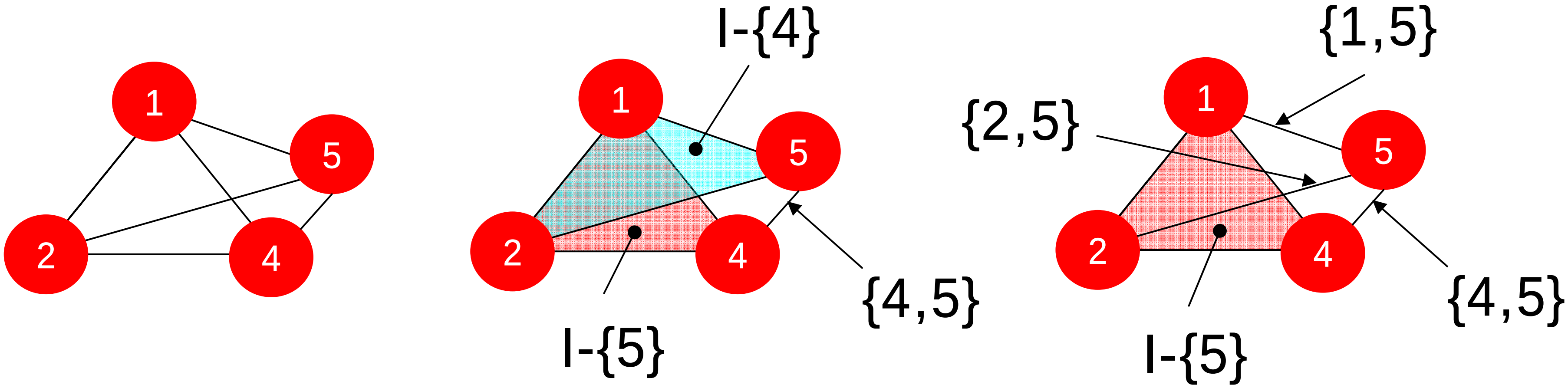
$$\begin{aligned}
\mathcal{J}_1(I) &= \bigotimes_{\forall i \in I} \mathcal{J}_1(\{i\}) \\
\mathcal{J}_2(I) &= \bigotimes_{\forall E \in \text{cover}(I)} \mathcal{J}_2(E) \\
\mathcal{J}_3(I) &= \bigotimes_{\forall \{x,y\} \subseteq I} \mathcal{J}_3(\{x,y\})
\end{aligned}$$

Optimizations

1. Dynamic programming
   - The optimal substructure appears in the theorem due to homogeneousness
2. Implement the operator ✸ using ZBDDs

Dynamic Programming

- Type 3, I = {1,2,4,5}



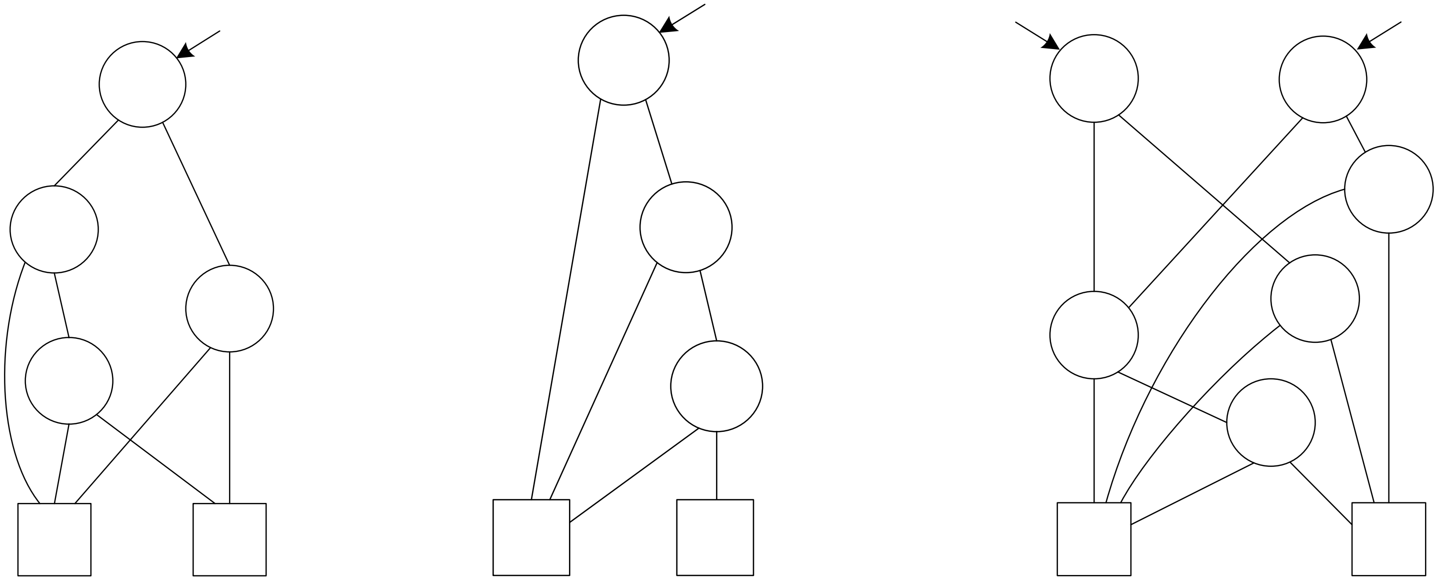
- $\mathcal{J}_3(\{1,2,4,5\}) = \mathcal{J}_3(\{1,2\}) \otimes \mathcal{J}_3(\{1,4\}) \otimes \mathcal{J}_3(\{1,5\}) \otimes \mathcal{J}_3(\{2,4\}) \otimes \mathcal{J}_3(\{2,5\}) \otimes \mathcal{J}_3(\{4,5\})$

- $\mathcal{J}_3(\{1,2,4,5\}) = \mathcal{J}_3(\{1,2,4\}) \otimes \mathcal{J}_3(\{1,2,5\}) \otimes \mathcal{J}_3(\{4,5\})$

- $\mathcal{J}_3(\{1,2,4,5\}) = \mathcal{J}_3(\{1,2,4\}) \otimes \mathcal{J}_3(\{1,5\}) \otimes \mathcal{J}_3(\{2,5\}) \otimes \mathcal{J}_3(\{4,5\})$

Efficient Implementation of ✸

- Use ZBDD
  - First, represent operands in ZBDD
  - Second, manipulate them directly in ZBDD

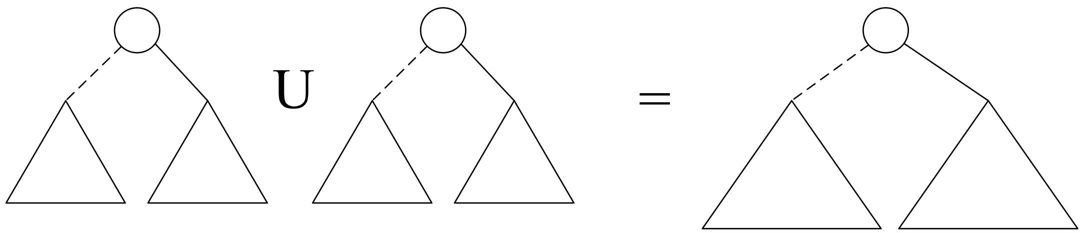


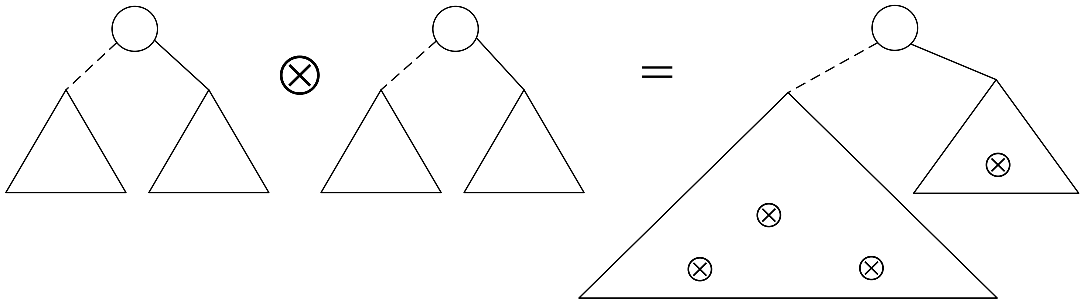
$\{\{3,4,5\}\}$

$\{\{1,4\},\{3,5\}\}$

Direct manipulation of ZBDD

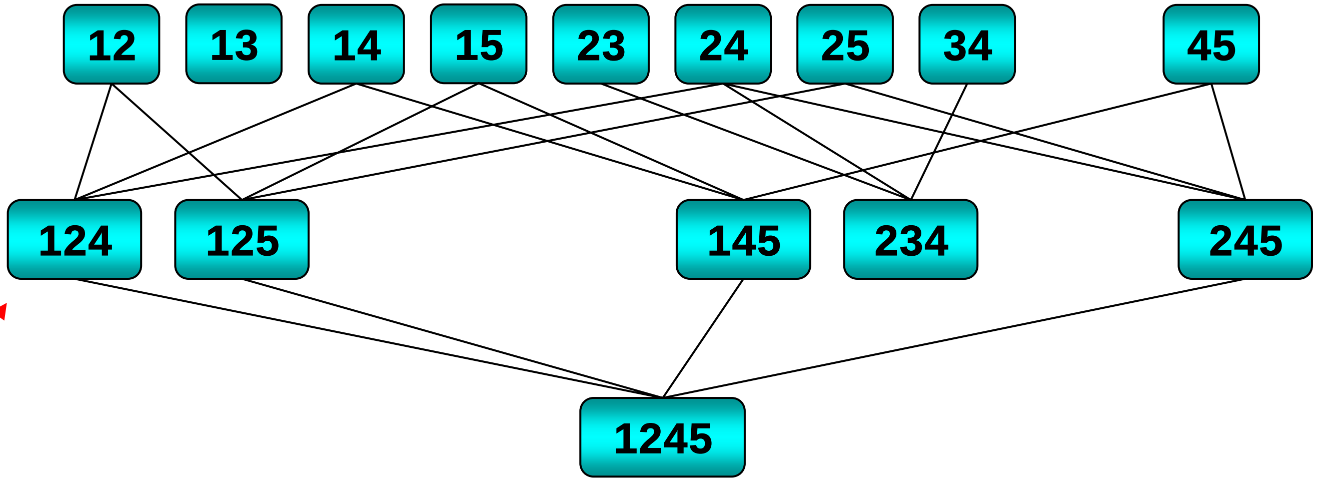
- Idea: recursive decomposition
  - If P = P_0 ☯ P_1 and Q = Q_0 ☯ Q_1
  - Then P ☯ Q = (P_0 ☯ Q_0) ☯ (P_1 ☯ Q_1)





x x

Finding Biclusters



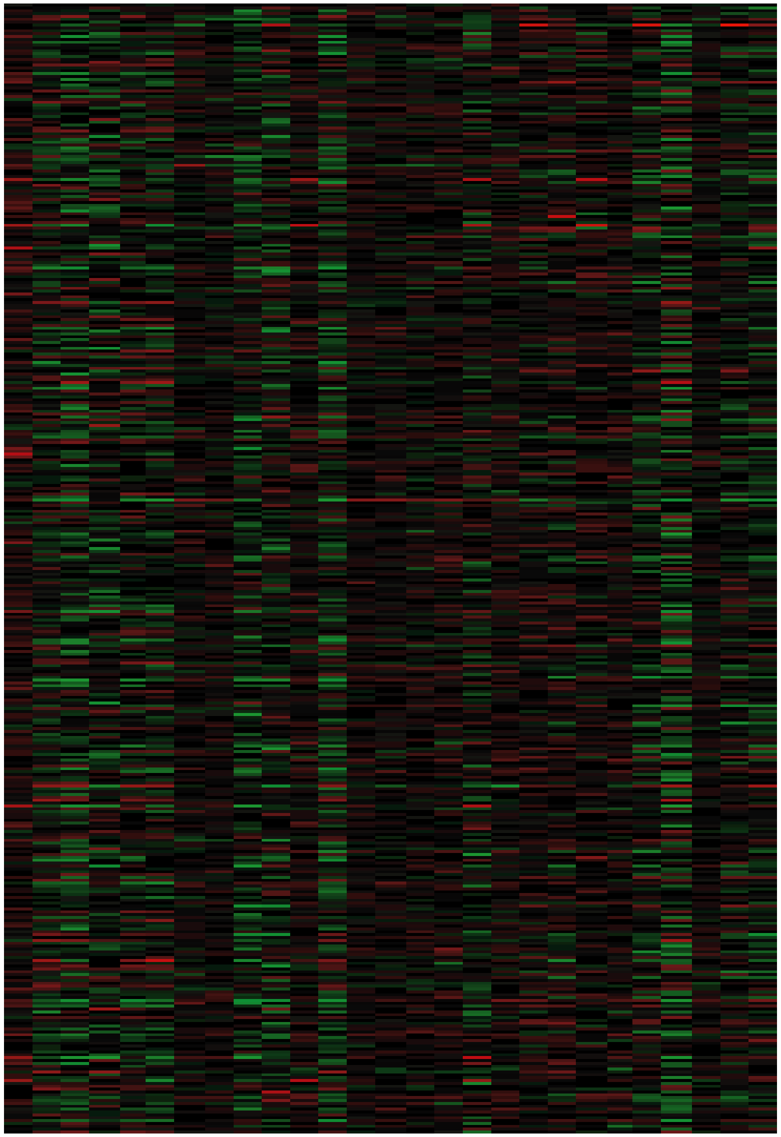
$$\mathcal{J}_3(\{1,2,4\}) = \mathcal{J}_3(\{1,2\})\otimes\mathcal{J}_3(\{1,4\})\otimes\mathcal{J}_3(\{2,4\})$$

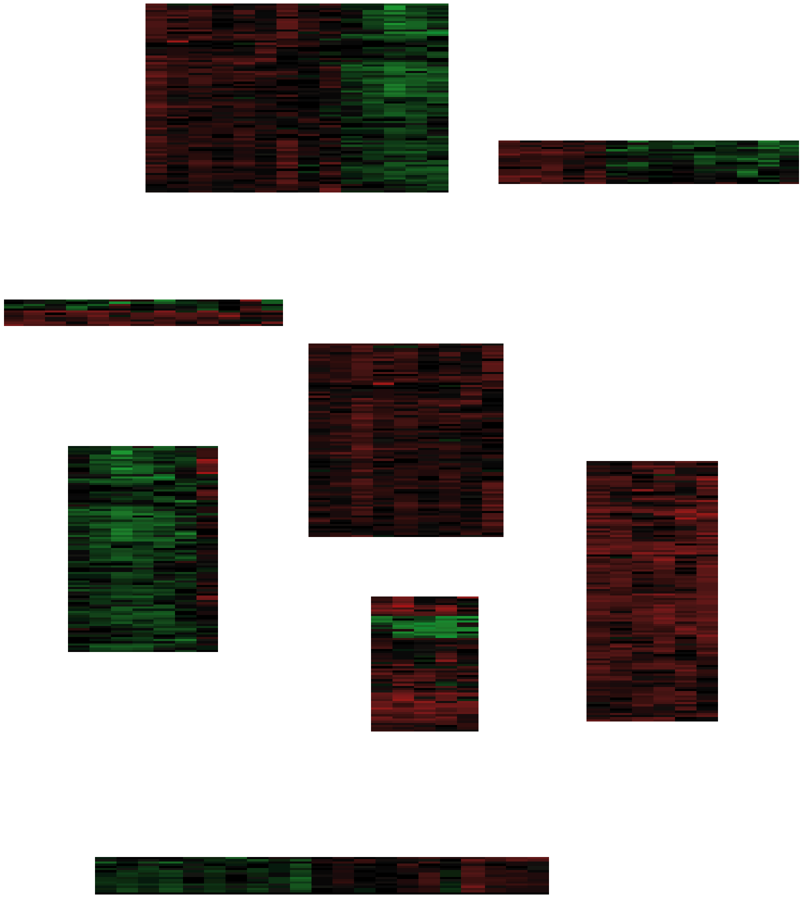
$$\mathcal{J}_3(\{1,2,4,5\}) = \mathcal{J}_3(\{1,2,4\})\otimes\mathcal{J}_3(\{1,2,5\})\otimes\mathcal{J}_3(\{4,5\})$$

Outline

- Introduction
- Problem formulation
- Algorithm description
- Applications
  - DNA microarray analysis
  - Finding regulatory modules
  - Correlating genes with clinical parameters

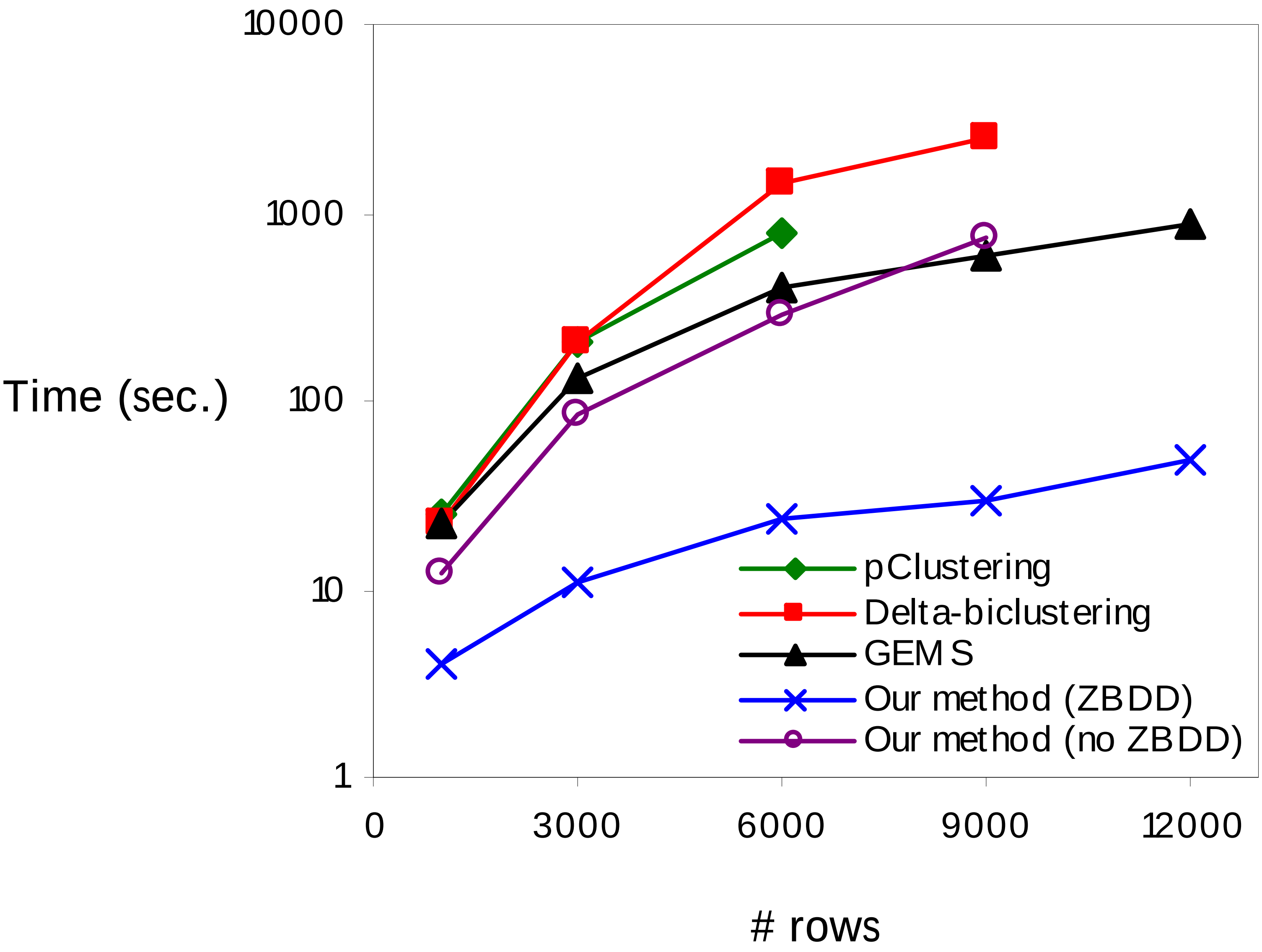
DNA Microarray



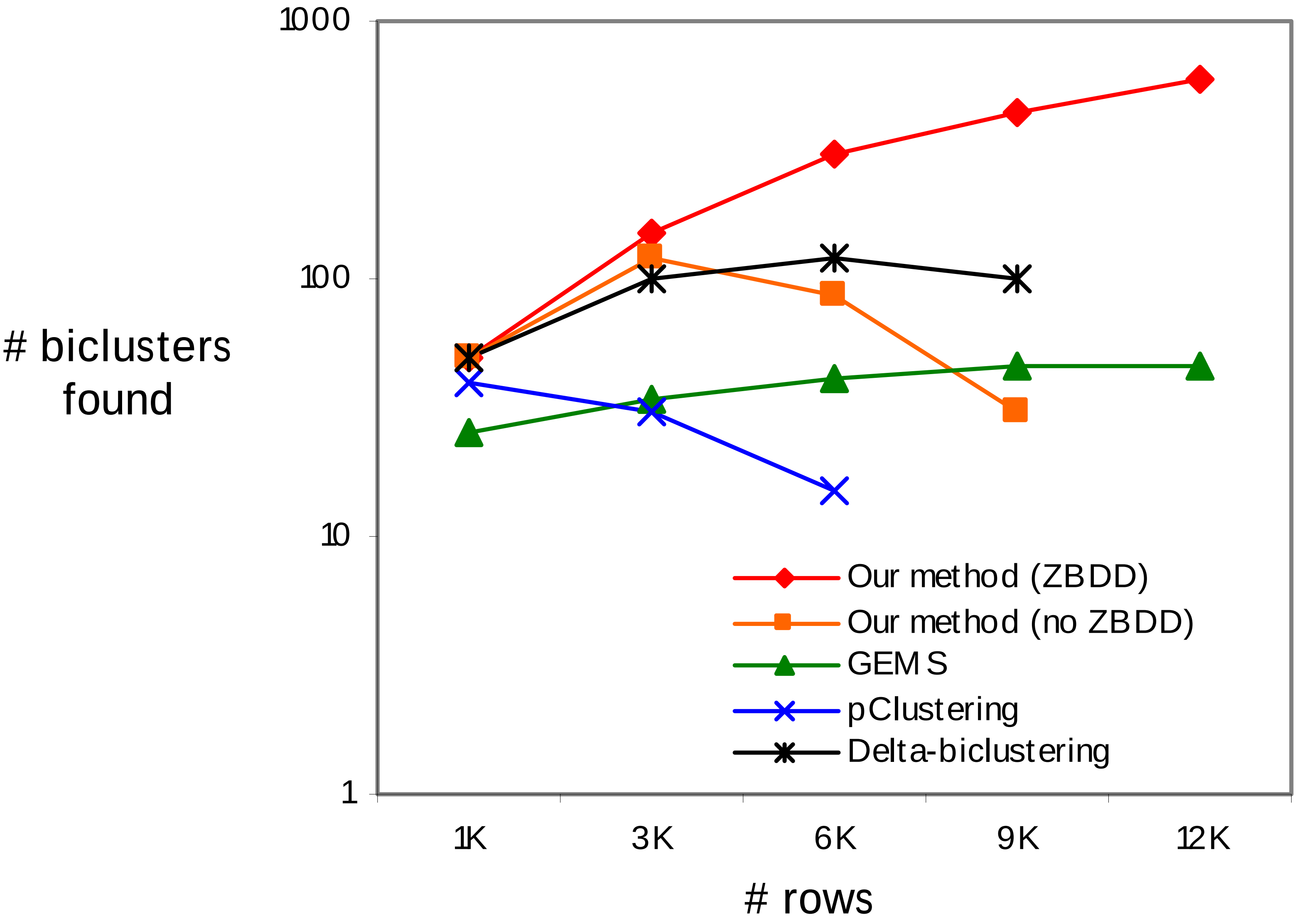


Renal cell carcinoma data (Higgins et al., 2003)

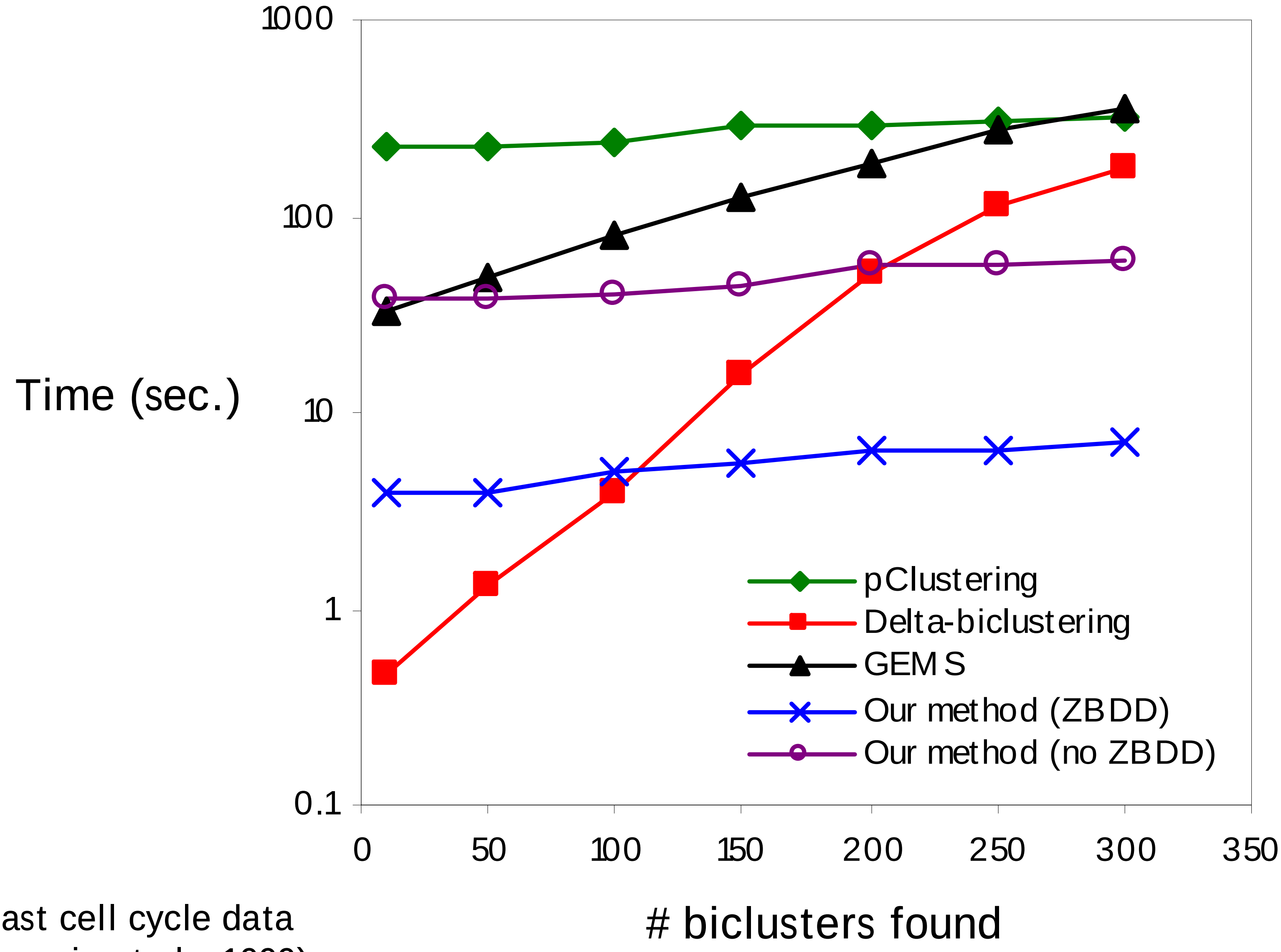
Synthetic Data



Synthetic Data

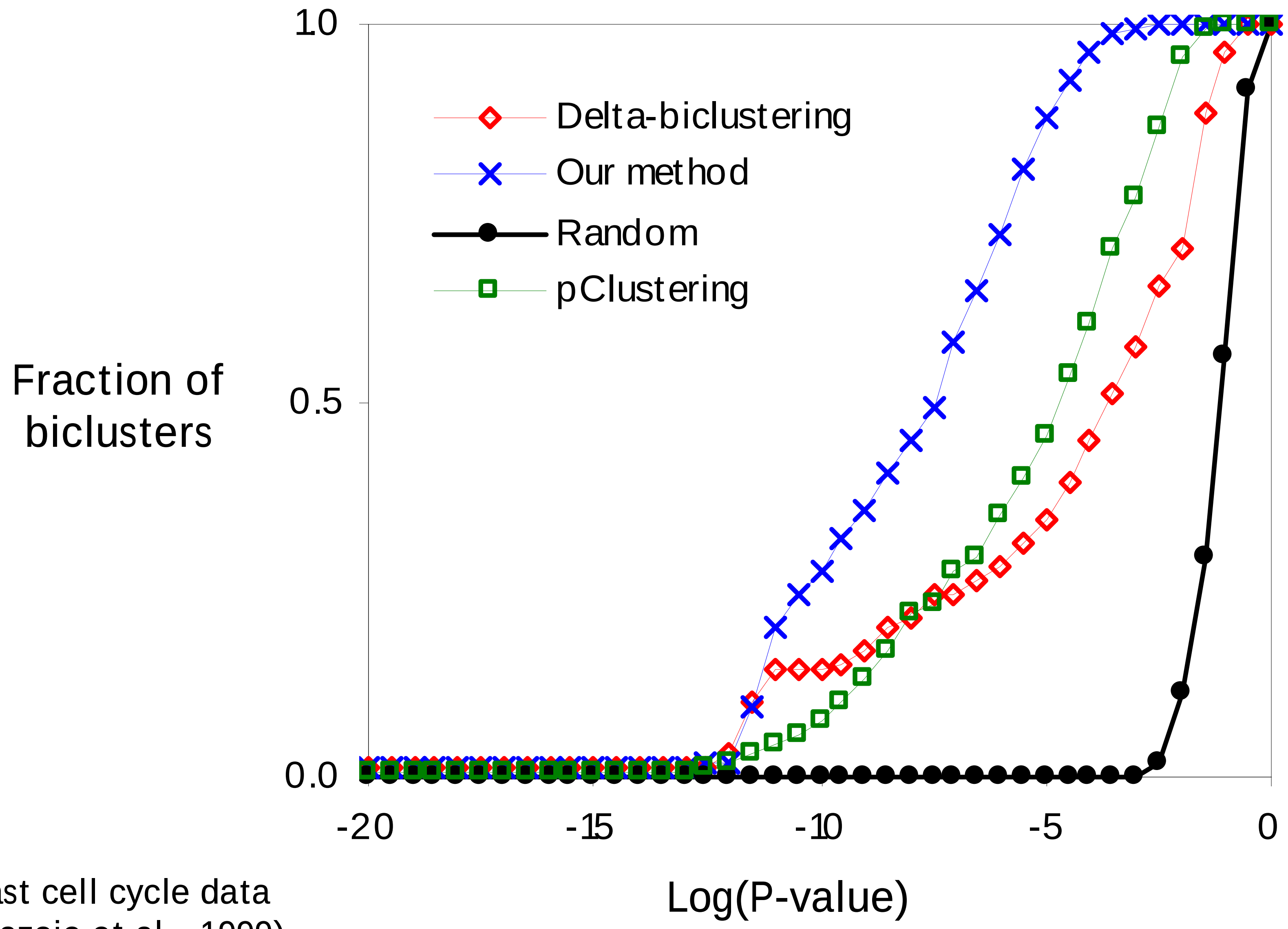


Actual Gene Expression Data



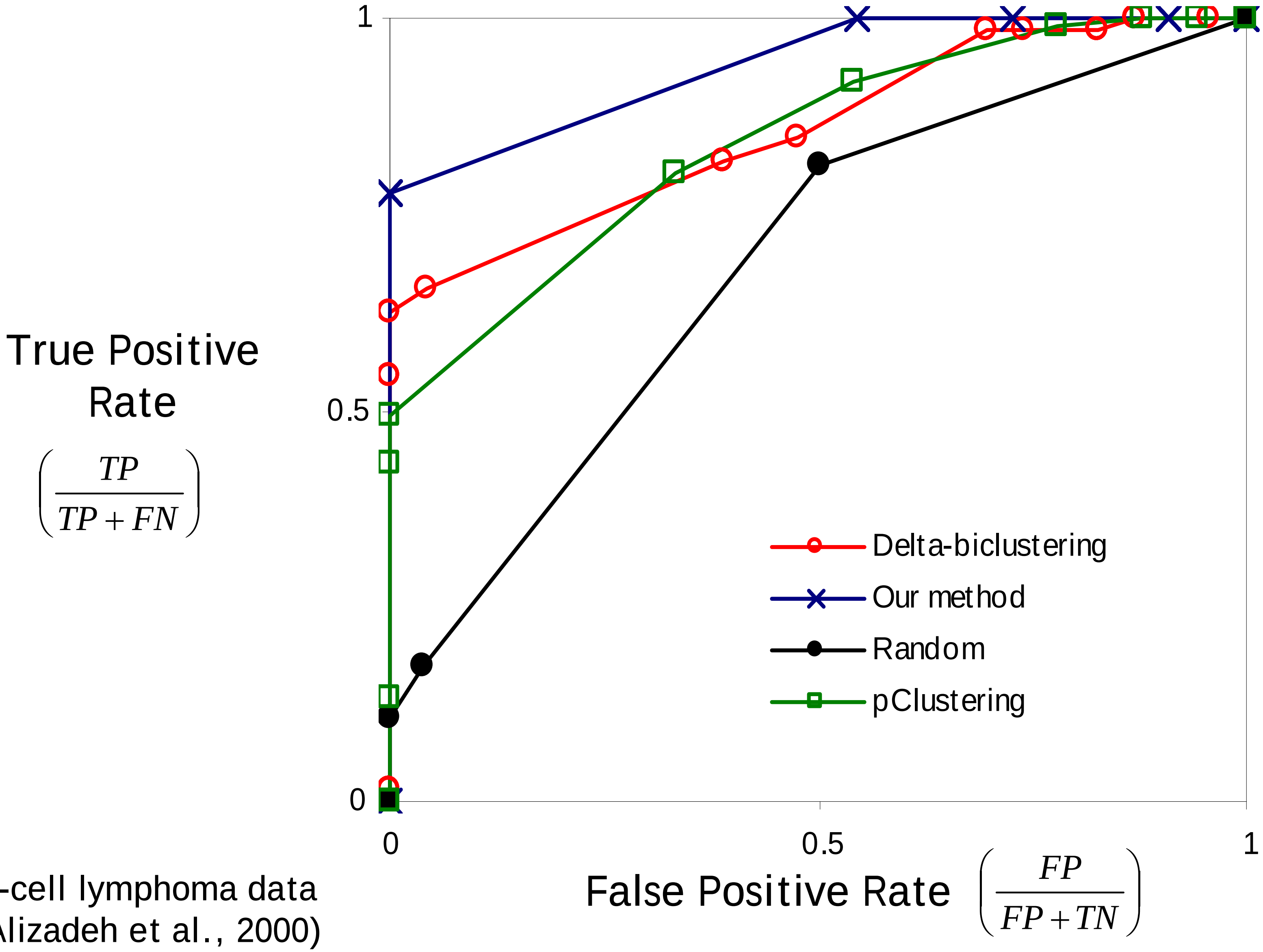
Yeast cell cycle data
(Tavazoie et al., 1999)

P-value



Yeast cell cycle data
(Tavazoie et al., 1999)

ROC Curve



B-cell lymphoma data
(Alizadeh et al., 2000)

Summary

- Explosion of genomic data
  - New techniques are emerging
- Biclustering
  - Useful but can be computationally expensive
- BDD
  - Efficient data structure + algorithms for BF
- Our biclustering algorithm
  - Leveraged by ZBDD
  - Exact and scalable algorithm
- Various applications

For Further Information

- S. Yoon, C. Nardini, L. Benini and G. De Micheli. “Discovering Coherent Biclusters from Gene Expression Data Using Zero-suppressed Binary Decision Diagrams,” *IEEE Transactions on Computational Biology and Bioinformatics*, to appear.