Biological Network Analysis: Graph Mining in Bioinformatics

Karsten Borgwardt

Interdepartmental Bioinformatics Group
MPIs Tübingen

with permission from Xifeng Yan and Xianghong Jasmine Zhou

Mining coherent dense subgraphs across massive biological networks for functional discovery

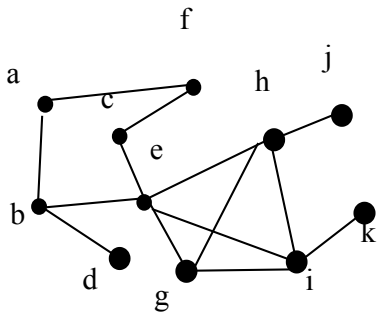
H. Hu[1], X. Yan[2], Y. Huang[1], J. Han[2], and X. J. Zhou[1]

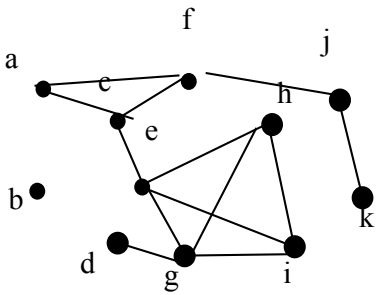
[1]University of Southern California
[2]University of Illinois at Urbana-Champaign

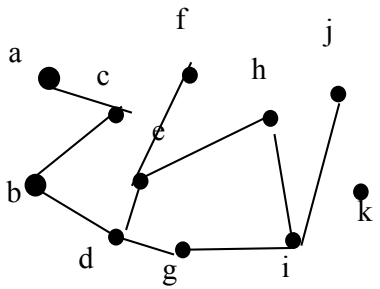
Biological Networks

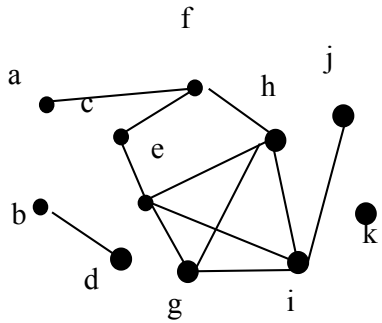
- Protein-protein interaction network
- Metabolic network
- Transcriptional regulatory network
- Co-expression network
- Genetic Interaction network
- …

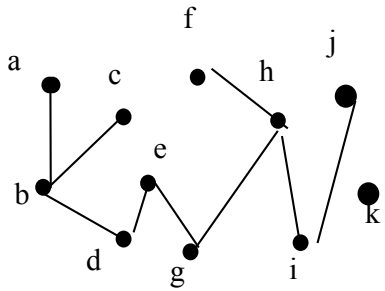
Data Mining Across Multiple Networks

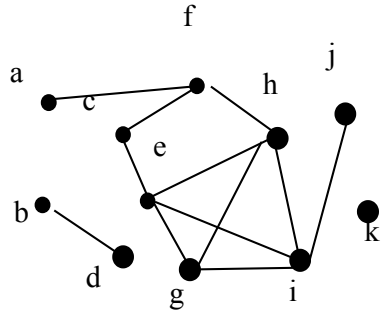




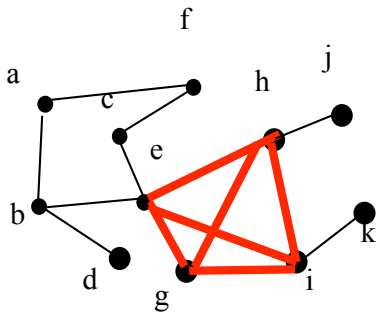


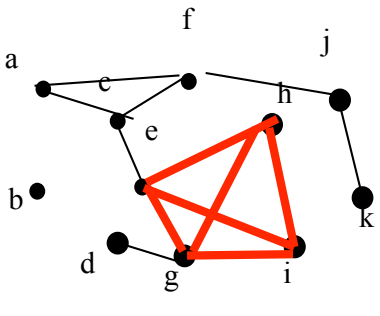


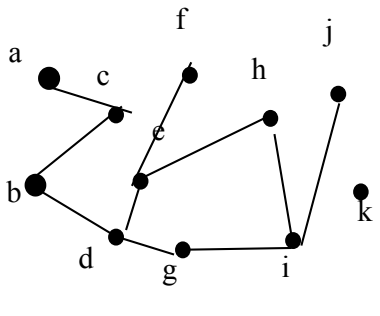


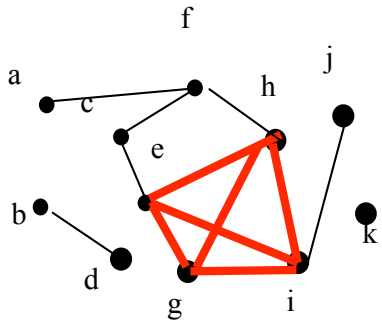


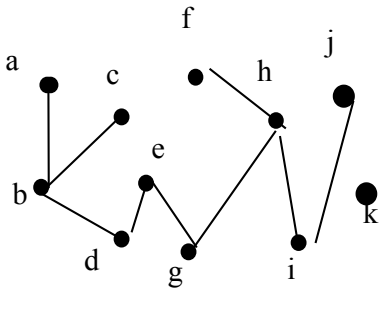
Data Mining Across Multiple Networks

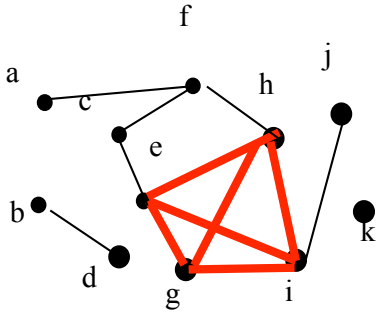




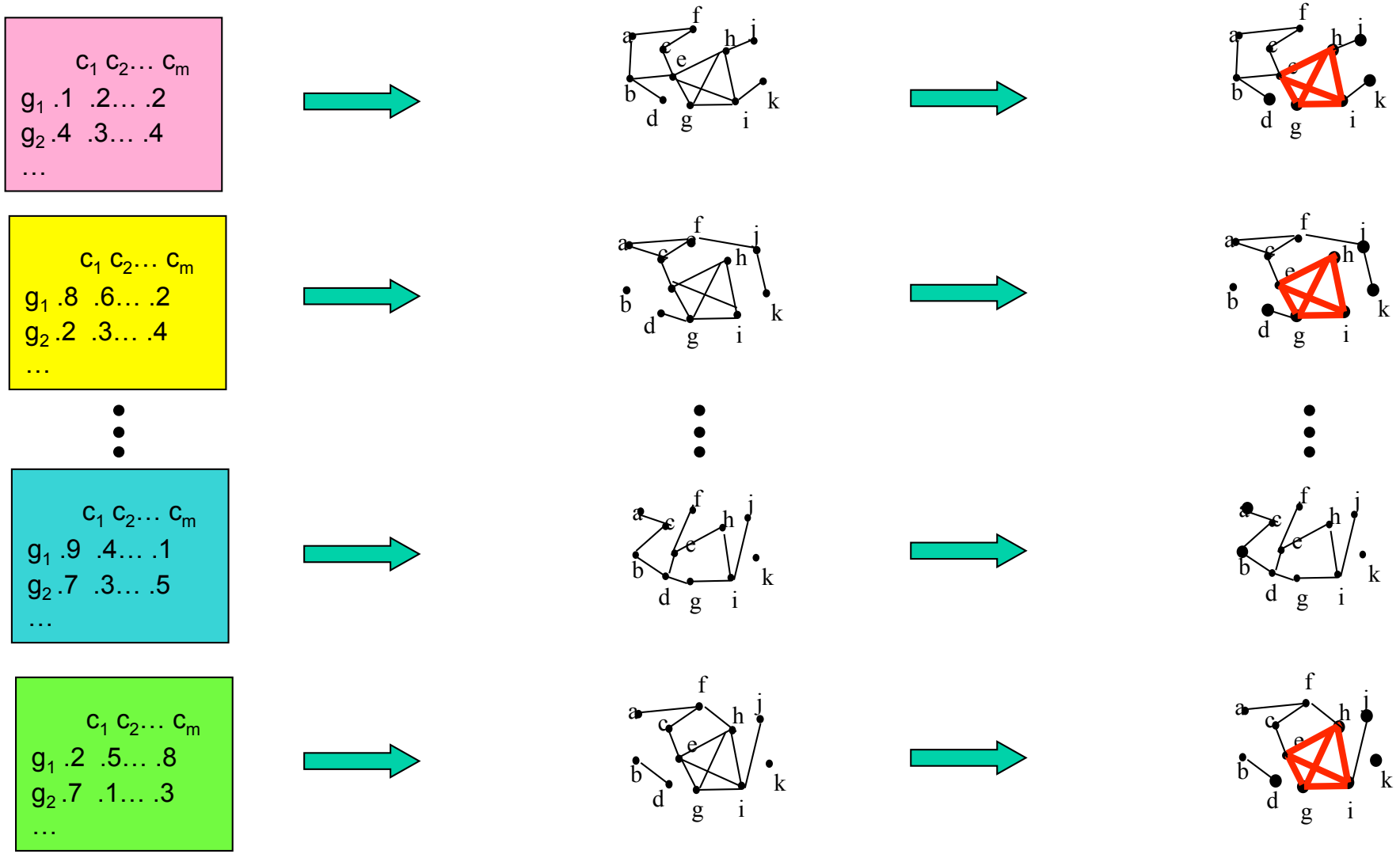








Identify frequent co-expression clusters across multiple microarray data sets



Frequent Subgraph Mining Problem is hard!

Problem formulation: Given *n* graphs, identify subgraphs which occur in at least *m* graphs (m ≤ *n*)

Efficient modeling of Biological Networks: each gene occurs once and only once in a graph. That means, the edge labels are unique.

The common pattern growth approach

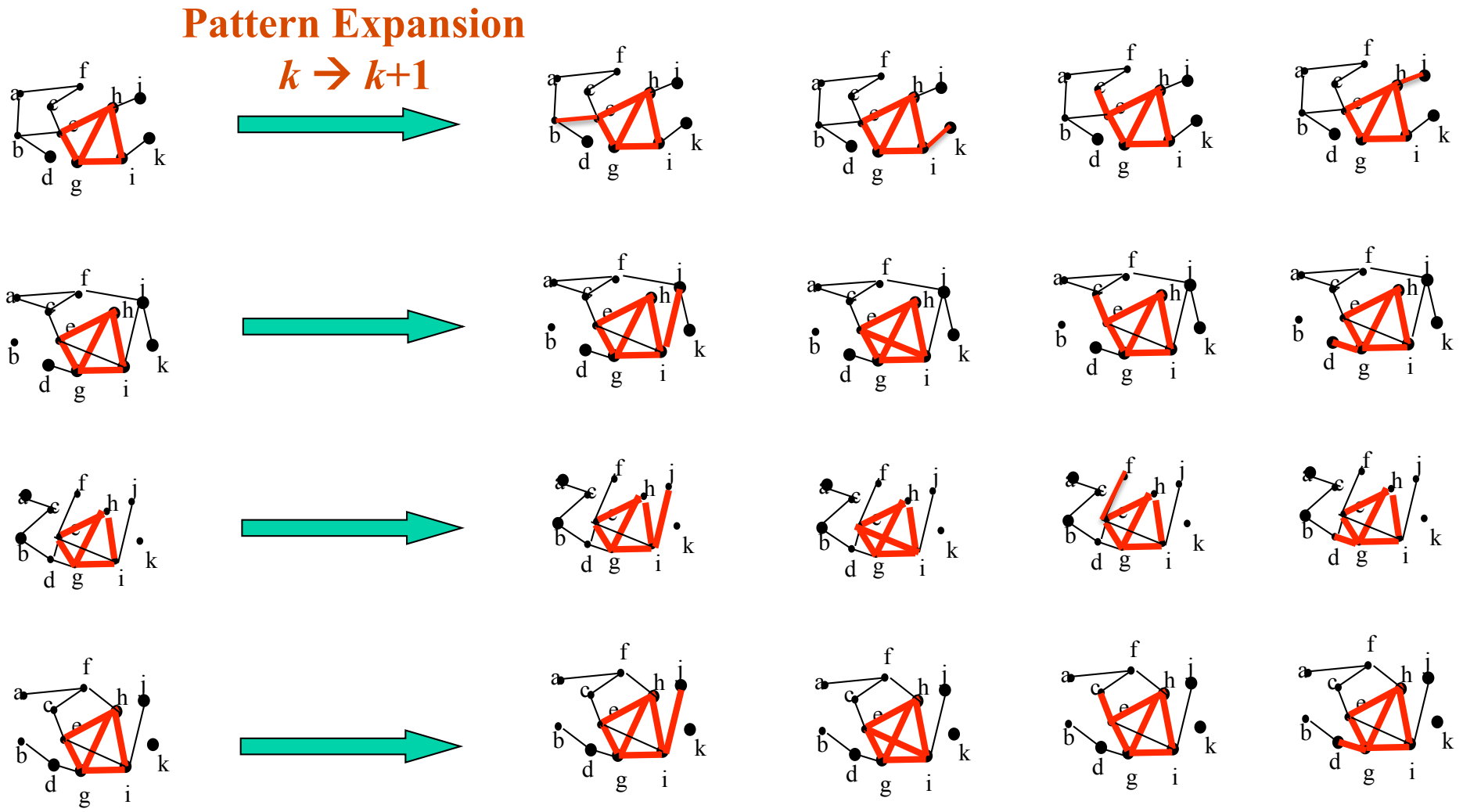
Find a frequent subgraph of *k* edges, and expand it to *k+1* edge to check occurrence frequency

- Koyuturk M., Grama A. & Szpankowski W. *An efficient algorithm for detecting frequent subgraphs in biological networks*. ISMB 2004
- Yan, Zhou, and Han. *Mining Closed Relational Graphs with Connectivity Constraints*. ICDE 2005

Problem of the Pattern-growth approach

The time and memory requirements increase exponentially with increasing size of patterns and increasing number of networks. The number of frequent dense subgraphs is explosive when there are very large frequent dense subgraphs, e.g., subgraphs with hundreds of edges.

Problem of the Pattern-growth approach



Our solution

We develop a novel algorithm, called *CODENSE*, to mine frequent ***coherent dense*** subgraphs. The target subgraphs have three characteristics:

(1) All edges occur in >= k graphs (frequency)

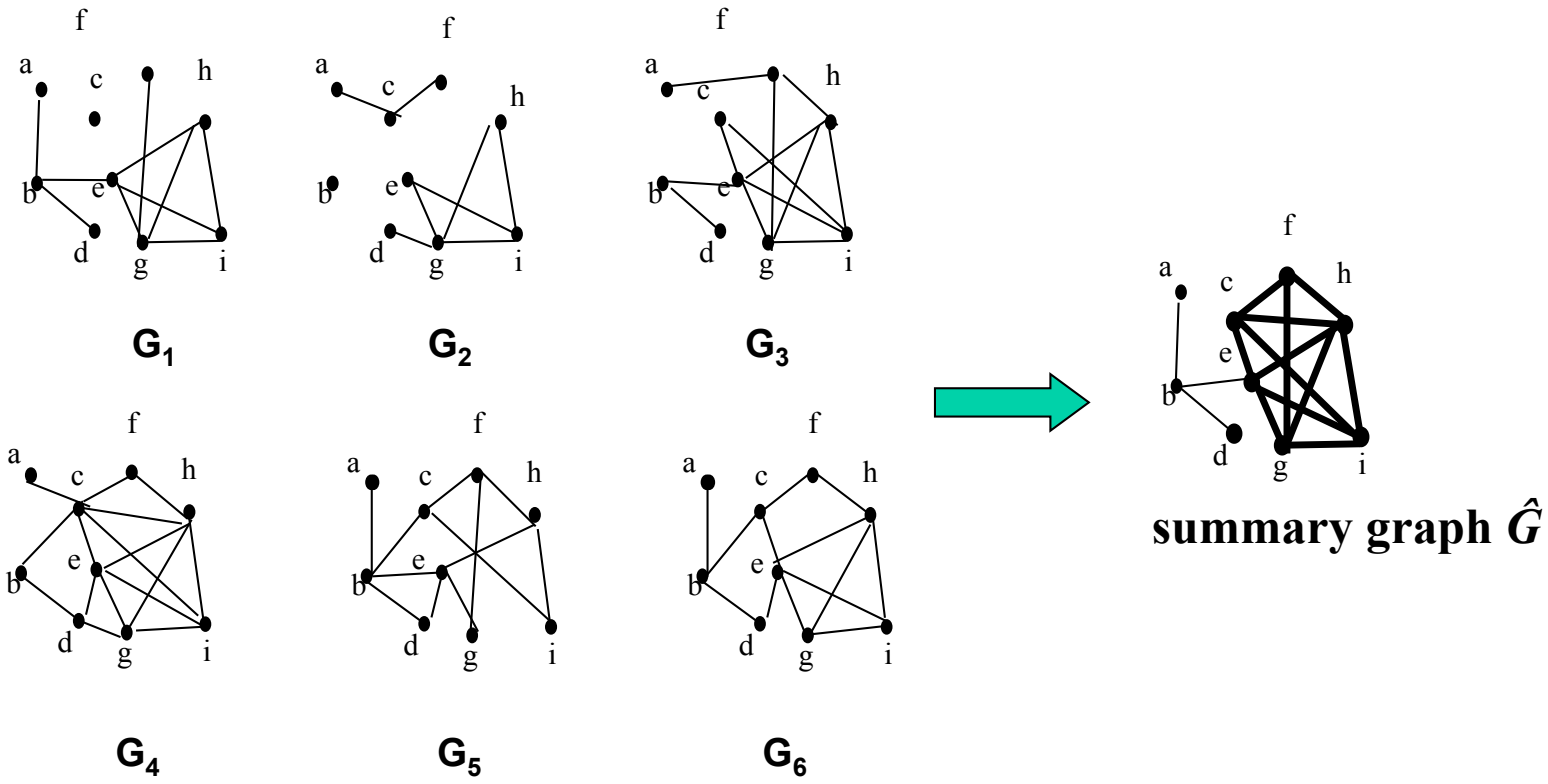
(2) All edges should exhibit correlated occurrences in the given graph set. (coherency)

(3) The subgraph is dense, where density *d* is higher than a threshold γ and *d=2m/(n(n-1))* (density)

m: #edges, n: #nodes

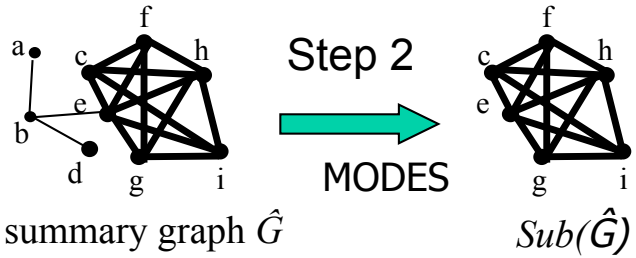
CODENSE: Mine coherent dense subgraph

(1) Builds a summary graph by eliminating infrequent edges



CODENSE: Mine coherent dense subgraph

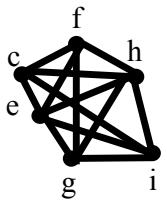
(2) Identify dense subgraphs of the summary graph



Observation: If a frequent subgraph is dense, it must be a dense subgraph in the summary graph. However, the reverse conclusion is not true.

CODENSE: Mine coherent dense subgraph

(3) Construct the edge occurrence profiles for each dense summary subgraph



$Sub(\hat{G})$

Step 3

E	G1	G2	G3	G4	G5	G6
c-e	0	0	1	1	0	1
c-f	0	1	0	1	1	1
c-h	0	0	0	1	1	1
c-i	0	0	1	1	1	0
e-f	0	0	0	1	1	1
…	…	…	…	…	…	…

edge occurrence profiles

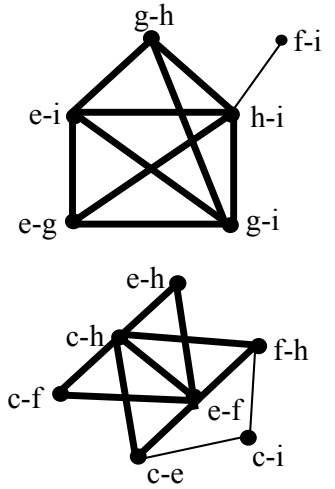
CODENSE: Mine coherent dense subgraph

(4) builds a second-order graph for each dense summary subgraph

E	G1	G2	G3	G4	G5	G6
c-e	0	0	1	1	1	1
c-f	0	1	0	1	1	1
c-h	0	0	0	1	1	1
c-i	0	0	1	1	1	0
e-f	0	0	0	1	1	1
…	…	…	…	…	…	…

edge occurrence profiles

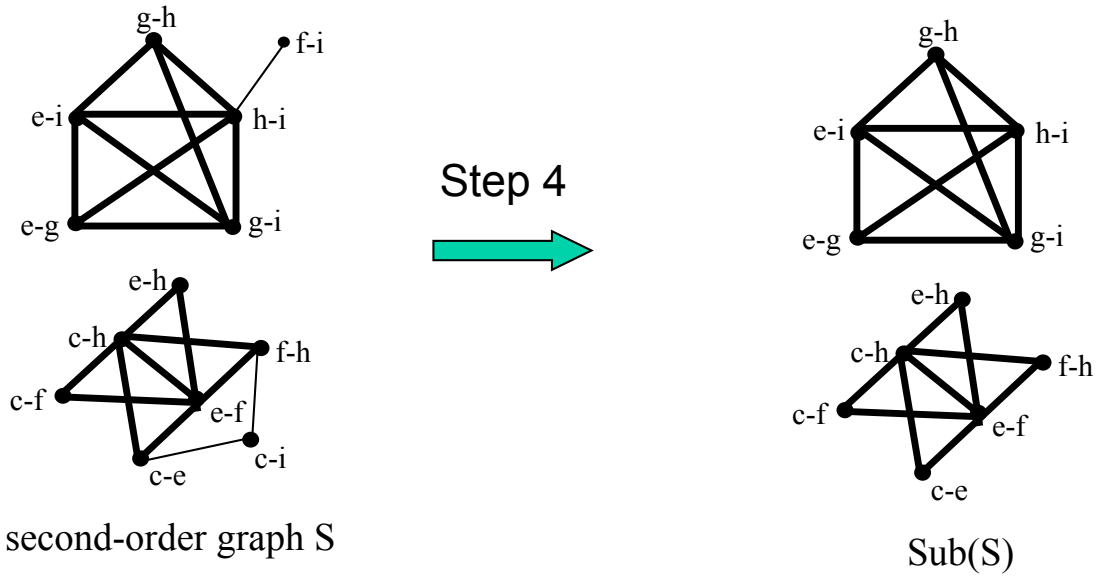
Step 4



second-order graph S

CODENSE: Mine coherent dense subgraph

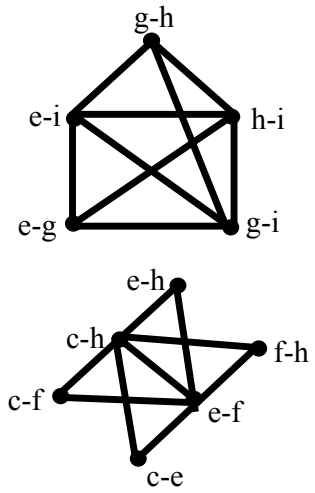
(5) Identify dense subgraphs of the second-order graph



Observation: if a subgraph is coherent (its edges show high correlation in their occurrences across a graph set), then its 2nd-order graph must be dense.

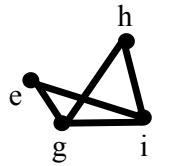
CODENSE: Mine coherent dense subgraph

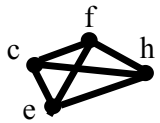
(6) Identify the coherent dense subgraphs



Sub(S)

Step 5





Sub(G)

Our solution

We develop a novel algorithm, called *CODENSE*, to mine frequent ***coherent dense*** subgraphs. The target subgraphs have three characteristics:

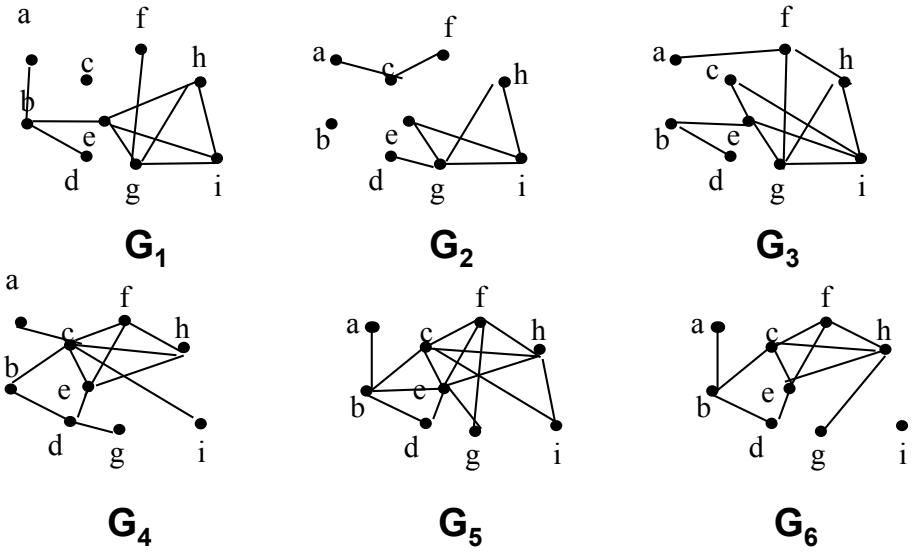
(1) All edges occur in >= k graphs (frequency)

(2) All edges should exhibit correlated occurrences in the given graph set. (coherency)

(3) The subgraph is dense, where density *d* is higher than a threshold γ and *d=2m/(n(n-1))* (density)

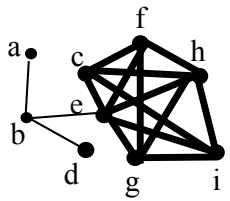
m: #edges, n: #nodes

CODENSE: Mine coherent dense subgraph



Step 1

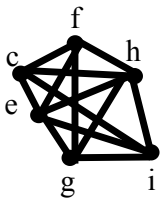
Add/Cut



summary graph $\hat{G}$

Step 2

MODES



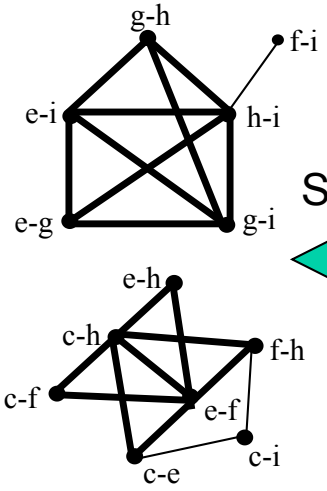
Sub($\hat{G}$)

Step 3

E	G1	G2	G3	G4	G5	G6
c-e	0	0	1	1	1	1
c-f	0	1	0	1	1	1
c-h	0	0	0	1	1	1
c-i	0	0	1	1	1	0
e-f	0	0	0	1	1	1
…	…	…	…	…	…	…

edge occurrence profiles

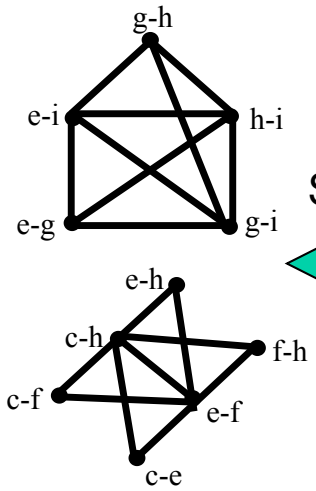
Step 4



second-order graph S

Step 5

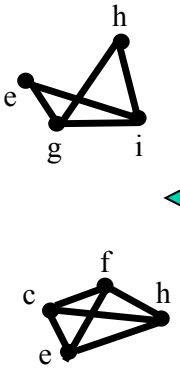
MODES



Sub(S)

Step 6

Restore G and MODES

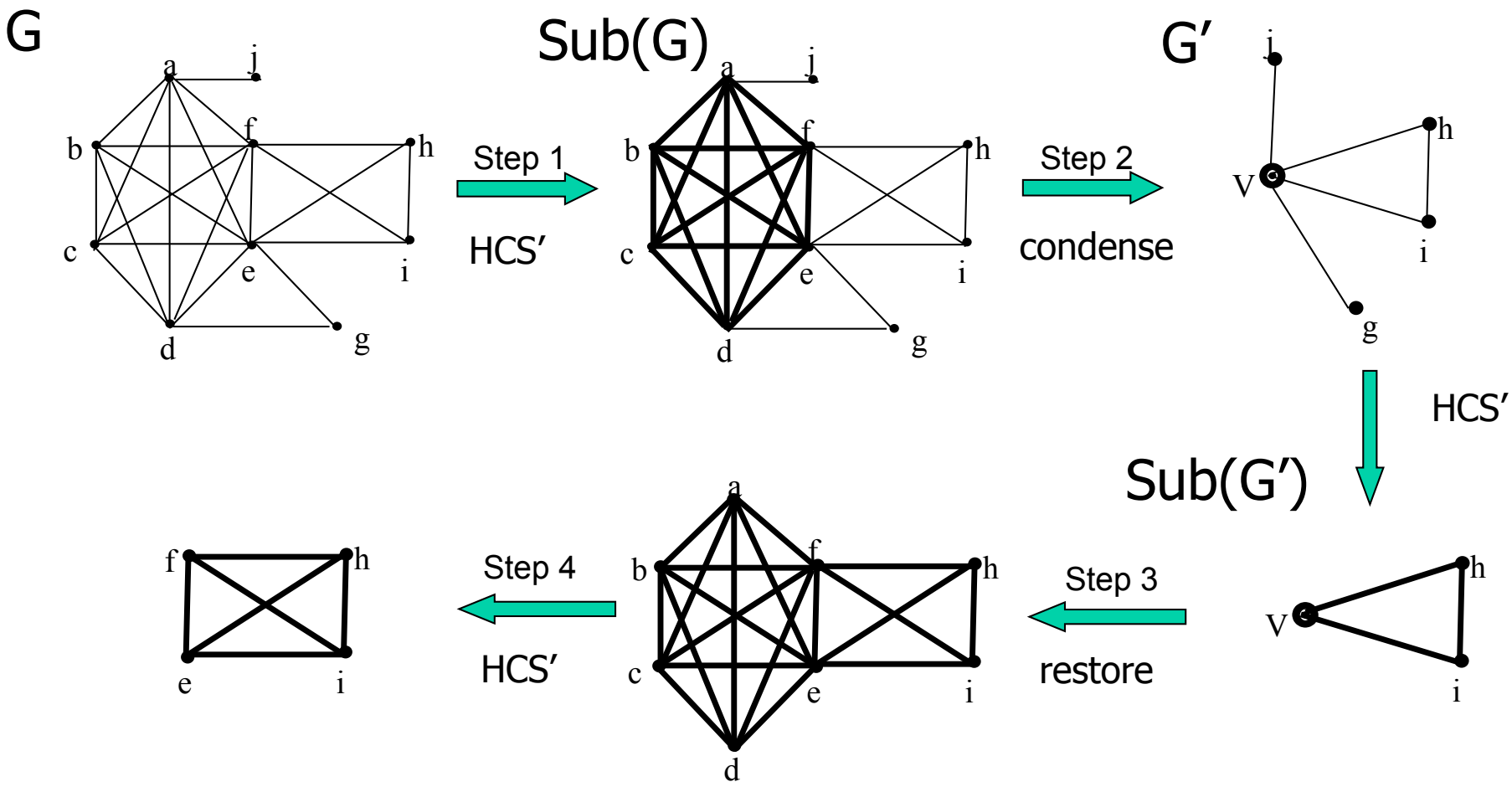


Sub(G)

CODENSE

The design of CODENSE can solve the scalability issue. Instead of mining each biological network individually, CODENSE compresses the networks into two meta-graphs and performs clustering in these two graphs only. Thus, CODENSE can handle any large number of networks.

MODES: Mine overlapped dense subgraph



Comparison with other Methods

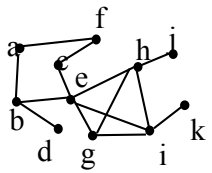
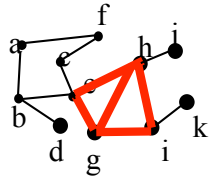
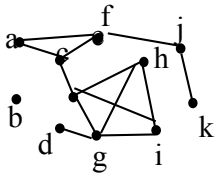
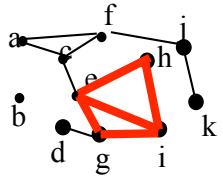
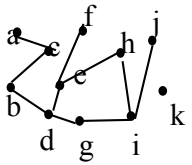
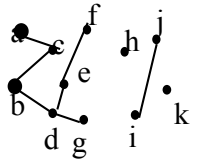
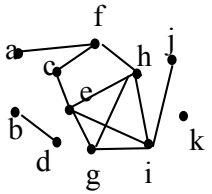
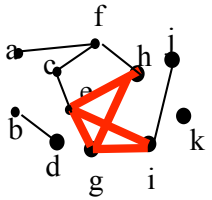
- By transforming all necessary information of the *n* graphs into two graphs, CODENSE achieves significant time and memory efficiency.

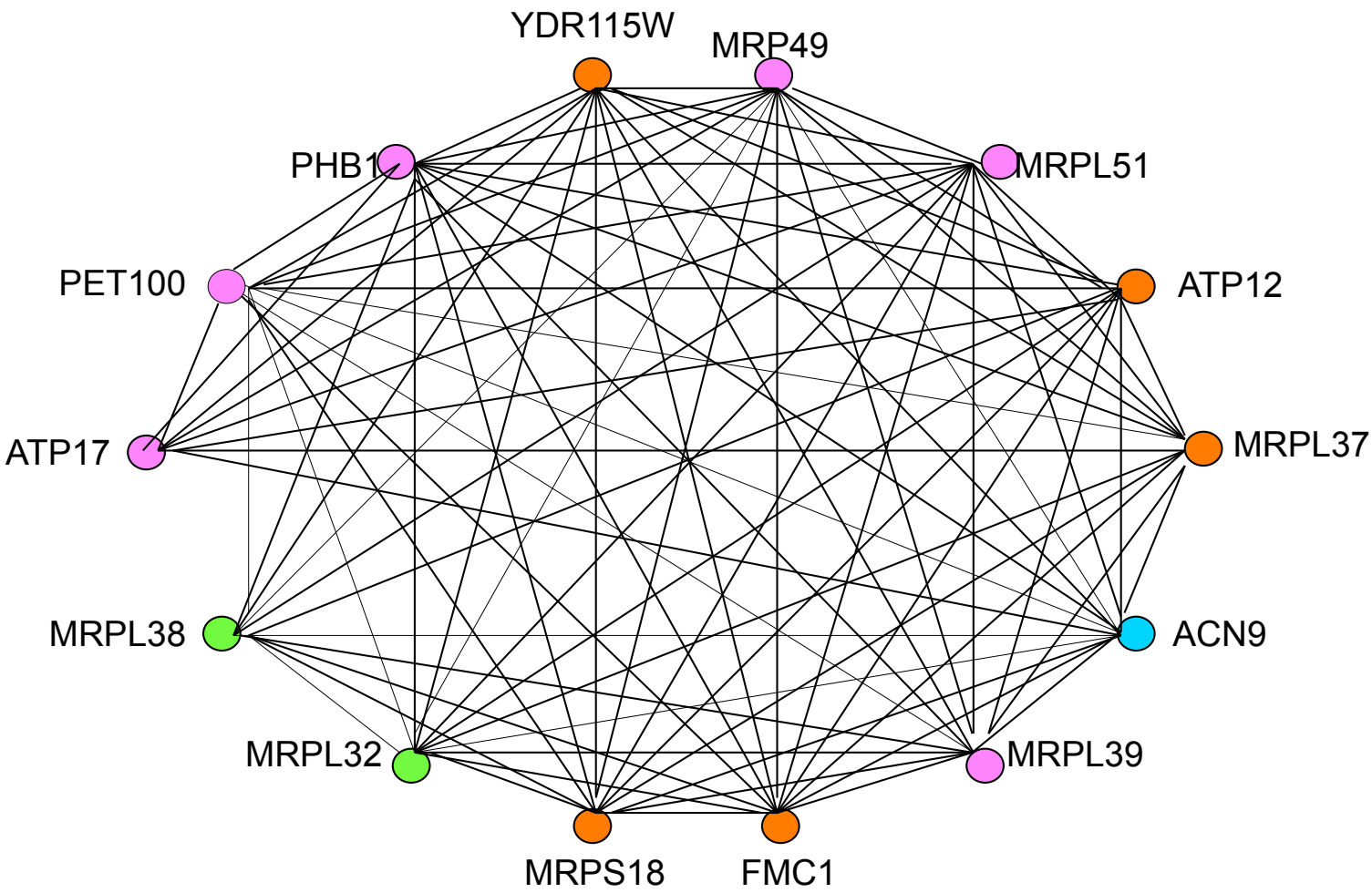
- CODENSE can mine both exact and approximate patterns.

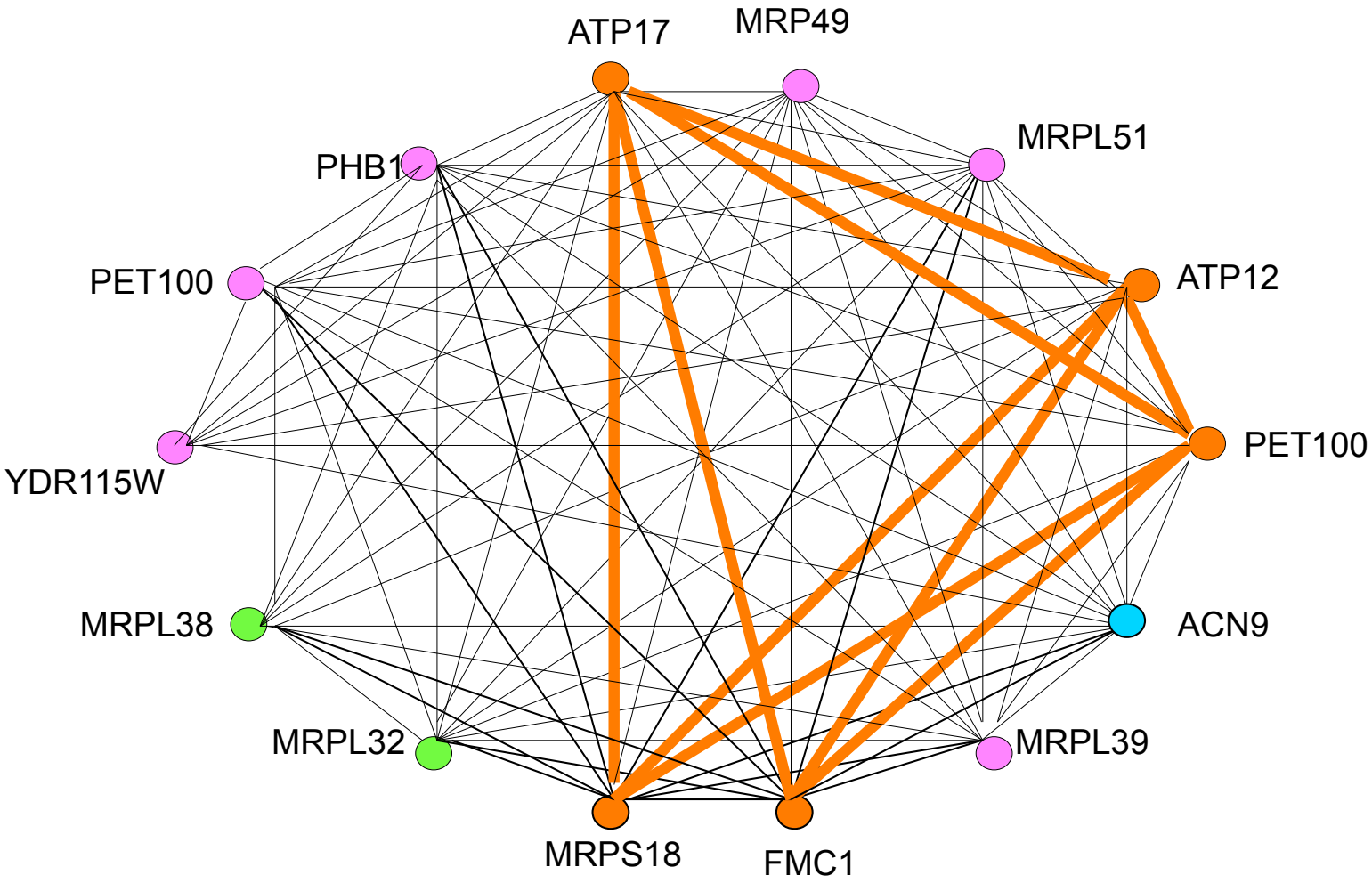
  (Approximate frequent subgraph mining is an important but never touched problem)

- CODENSE can be extended to pattern mining on weighted graphs

Applying CoDense to 39 yeast microarray data sets

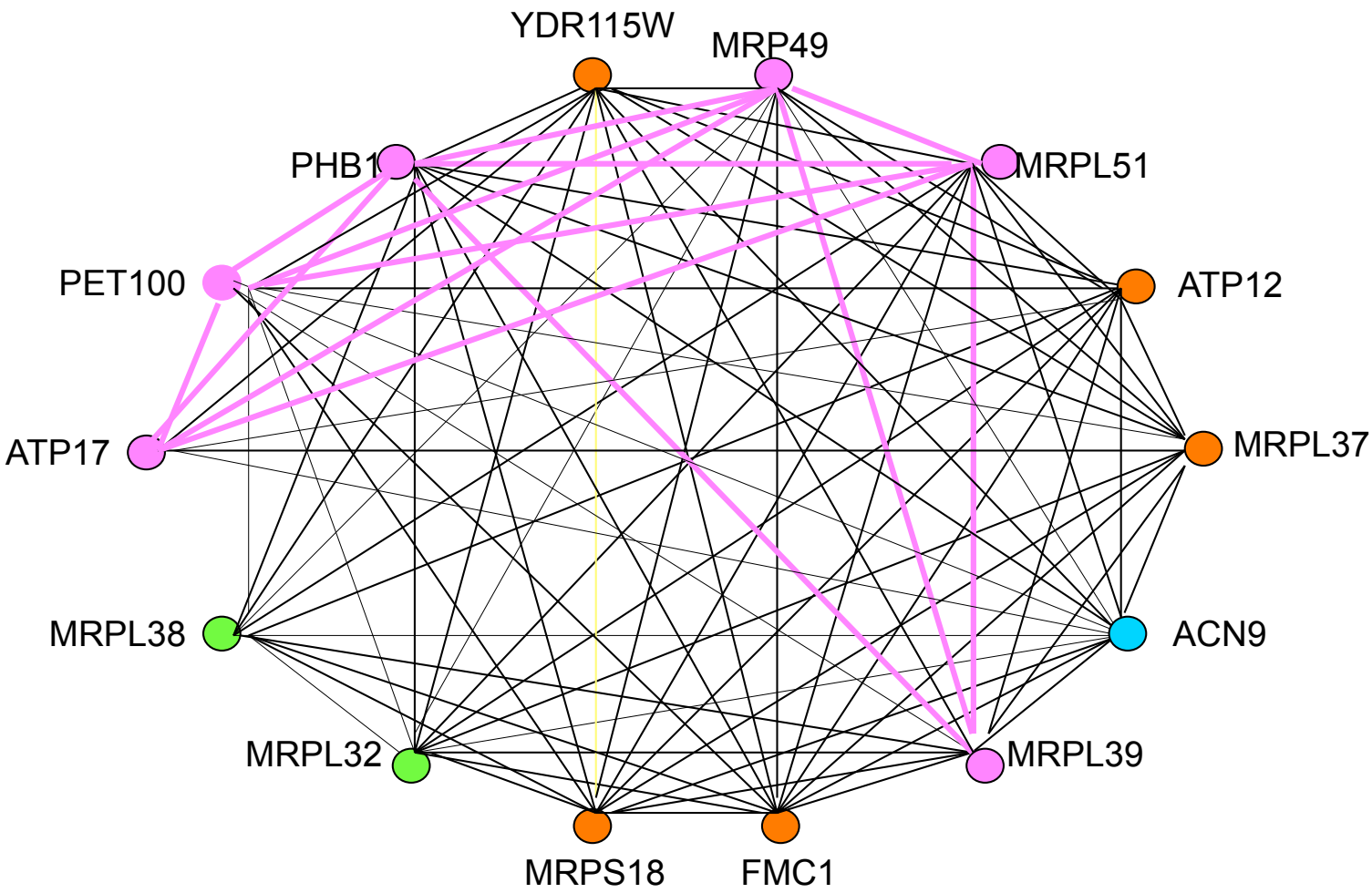
$c_1\ c_2 \ldots\ c_m$ g_1 .1 .2… .2 g_2 .4 .3… .4 …	→		→ 
$c_1\ c_2 \ldots\ c_m$ g_1 .8 .6… .2 g_2 .2 .3… .4 …	→		→ 
$c_1\ c_2 \ldots\ c_m$ g_1 .9 .4… .1 g_2 .7 .3… .5 …	→		→ 
$c_1\ c_2 \ldots\ c_m$ g_1 .2 .5… .8 g_2 .7 .1… .3 …	→		→ 


YDR115W
MRP49
PHB1
MRPL51
PET100
ATP12
ATP17
MRPL37
MRPL38
ACN9
MRPL32
MRPL39
MRPS18
FMC1



Yellow: YDR115W, FMC1, ATP12,MRPL37,MRPS18

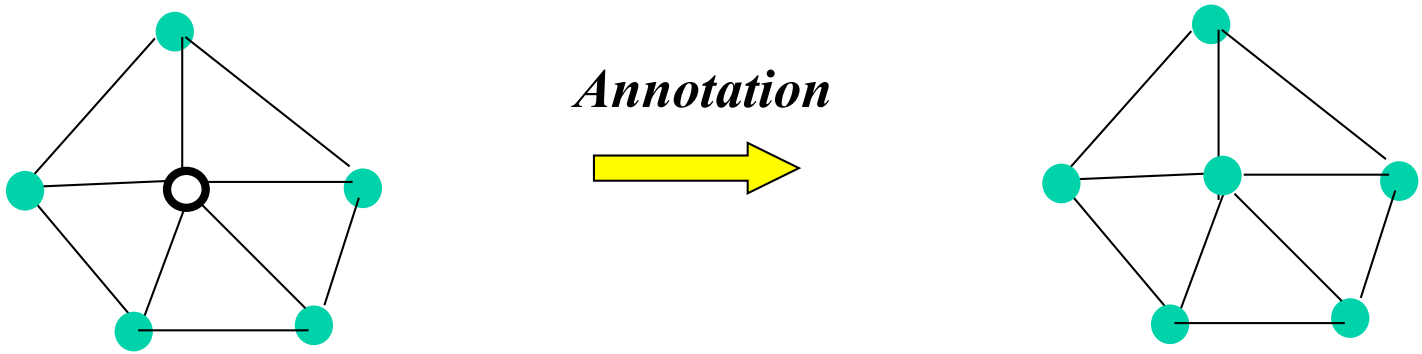
GO:0019538(protein metabolism; pvalue = 0.001122)



Red:PHB1,ATP17,MRPL51,MRPL39, MRPL49, MRPL51,PET100

GO:0006091(generation of precursor metabolites and energy; pvalue=0. 001339)

Functional annotation



Functional Annotation (Validation)

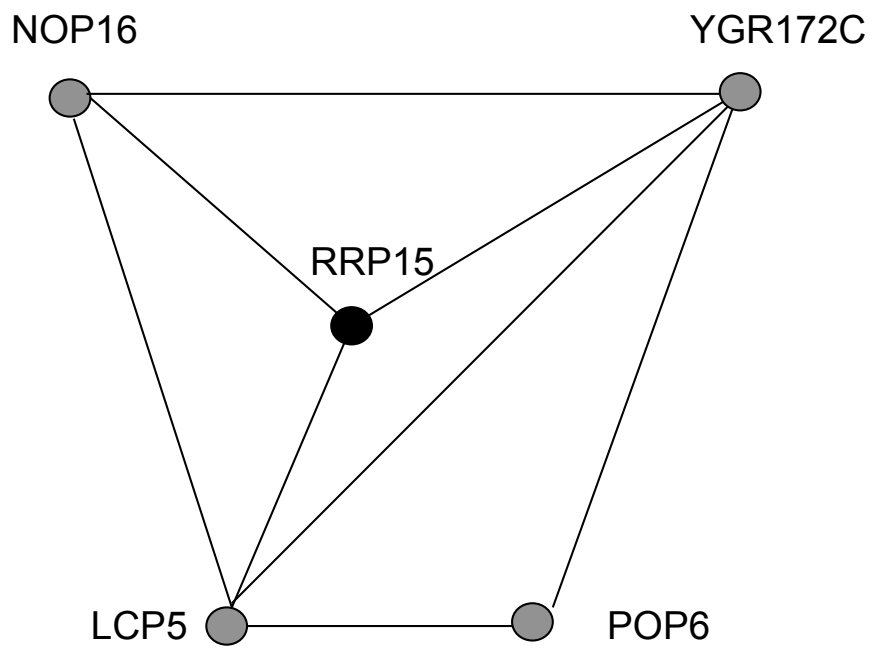
Method: leave-one-out approach - masking a known gene to be unknown, and assign its function based on the other genes in the subgraph pattern.

Functional categories: 166 functional categories at GO level at least 6

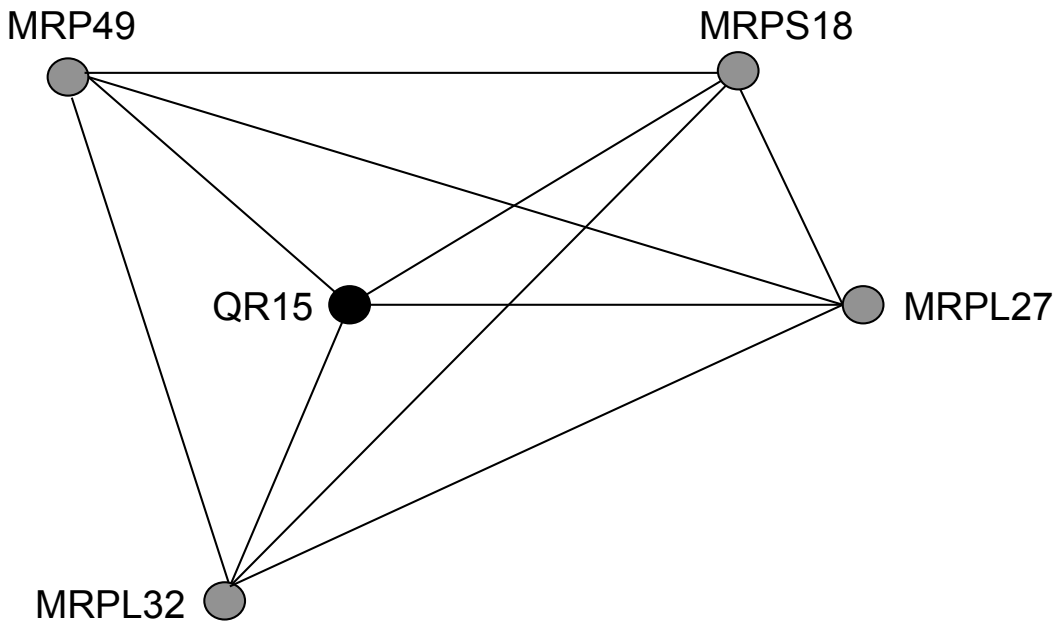
Results: 448 predictions with accuracy of 50%

Functional Annotation (Prediction)

We made functional predictions for 169 genes, covering a wide range of functional categories, e.g. amino acid biosynthesis, ATP biosynthesis, ribosome biogenesis, vitamin biosynthesis, etc. A significant number of our predictions can be supported by literature.



We predicted RRP15 to participate in "ribosome biogenesis". Based on a recent publication (De Marchis et al, RNA 2005), this gene is involved in pre-rRNA processing.



We predicted QRI5 to be involved in "protein biosynthesis"; QRI5 has been shown to participate in a common regulatory process together with MSS51 (Simon et al., 1992) and the GO annotation of MSS51 is "positive regulation of translation and protein biosynthesis".

Conclusion

- **We developed a scalable and efficient algorithm to mine coherent dense subgraphs across massive biological networks.**

- **It provides an efficient tool for the identification of network modules and for the functional discovery based on the biological network data.**

- **Our approach also provides a solution for cross-platform integration of microarray data.**

IBM

A graph-based approach to systematically reconstruct human transcriptional regulatory modules

Xifeng Yan*, Michael Mehan*, Yu Huang, Michael S. Waterman, Philip S. Yu, Xianghong Jasmine Zhou**

IBM T. J. Watson Research Center
University of Southern California

Rapid Accumulation of Microarray Data

- NCBI Gene Expression Omnibus

137231 experiments

- EBI Array Express

55228 experiments

The public microarray data increases by 3 folds per year

Microarray → Co-Expression Network

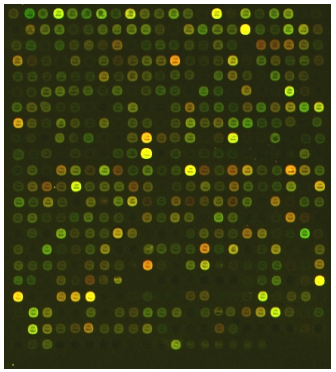
Microarray

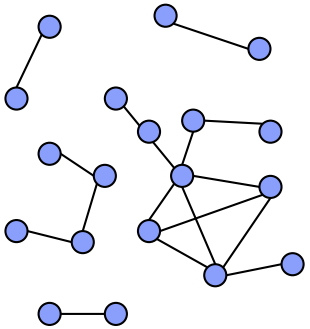
Coexpression Network

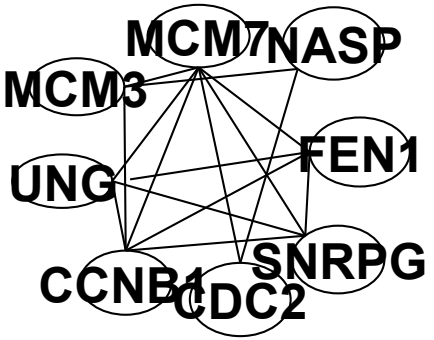
Module

conditions

genes







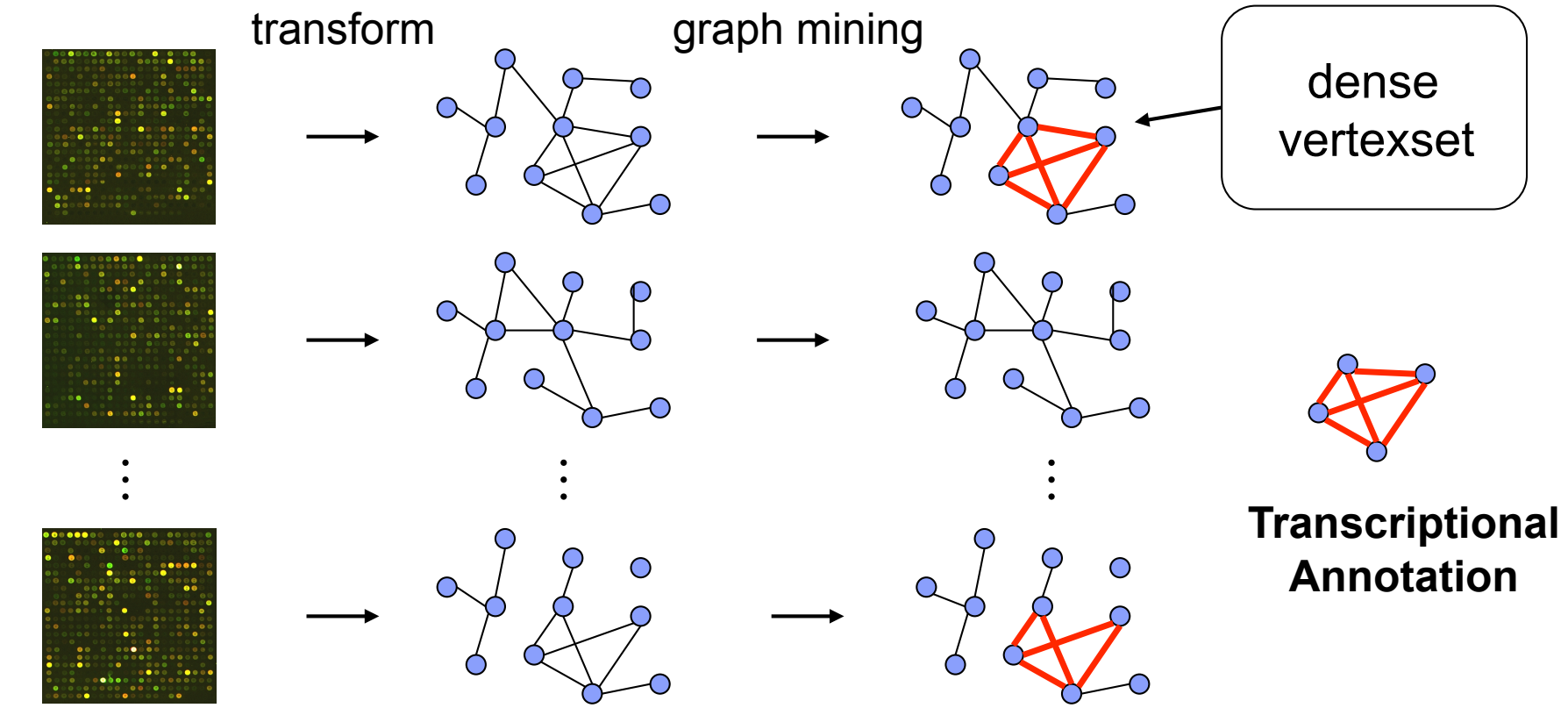
Two Issues:

- noise edges
- large scale

Solution: Single Graph → Multiple Graphs

Mining poor quality data!

Patterns discovered in multiple graphs are more reliable and significant



Frequent Dense Vertex Set

Given m networks, $G_1, G_2, \ldots, G_m$, $G_i = (V, E_i)$,

a density threshold δ, and

a frequency threshold θ,

A vertex subset M is a frequent dense pattern if

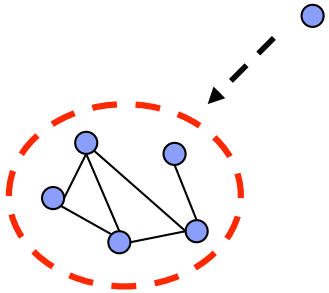
1. $M \subseteq V$,
2. among induced graphs $G_i[M]$, at least θm are dense,

$$\delta_i = \frac{2l_i}{n(n-1)} > \delta,$$

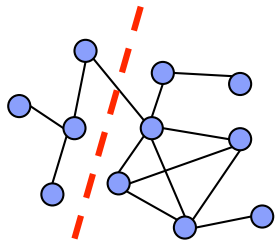
where l_i is # of edges of M in G_i and $n = |M|$.

Existing Solutions

- Bottom-up approach (small → large)
  - frequent maximum dense (KDD'05)



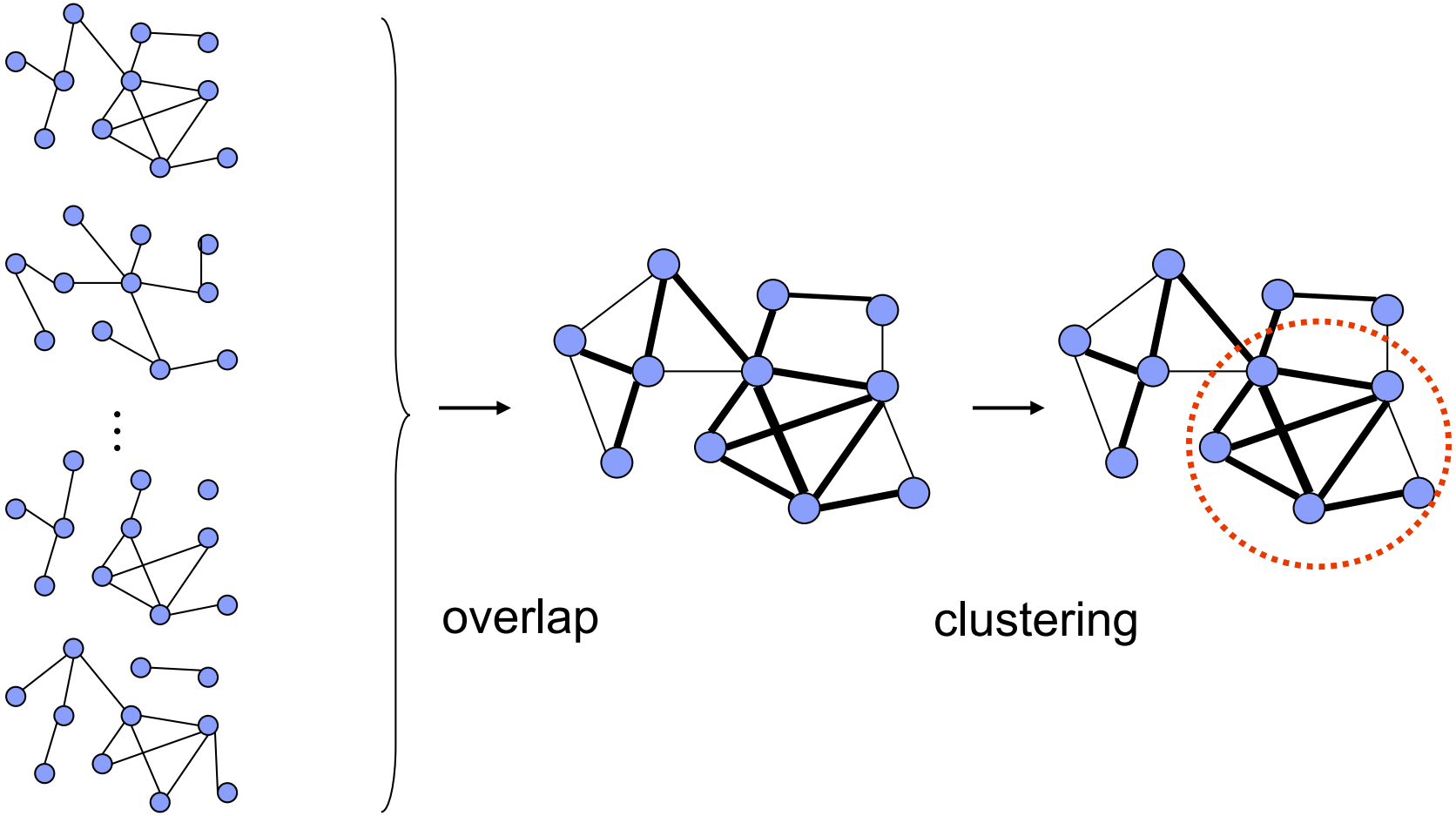
- Top-down approach (large → small)
  - consensus clustering (Filkov and Skiena 04)
  - summary graph (Lee etc. 04)



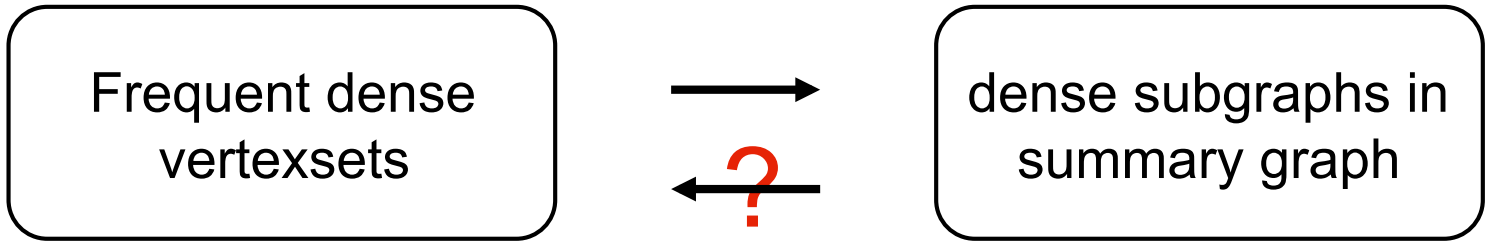
Our solutions

- Coherent clustering (Hu et al. ISMB'05)
- Partition and neighbor association (this work)

Summary Graph: Concept



Summary Graph: Noise Edges



- Dense subgraphs are accidentally formed by noise edges
- They are false frequent dense vertexsets
- Noise edges will also interfere with true modules

Summary Graph: Noise Edge Ratio

Assume noise edges occur i.i.d with probability q in individual network

Given m networks, the chance for a noise edge to have weight $\geq \theta m$ in a summary graph is

noise edge ratio in individual graph

$$b(m, \theta, q) = \sum_{l=\lceil \theta m \rceil}^{m} \binom{m}{l} q^l (1-q)^{m-l}. \qquad (1)$$

noise edge ratio in summary graph

In a typical setting where $m = 100, \theta m = 5$

$$q > 2\% \rightarrow b > 5\%$$

Summary Graph: False Patterns by Noise Edges

Overlay Model

$$G = G' + G^*$$

Observed noise real

Assume G' is a random graph, $G'(n, p)$.
Let $s = k(k-1)/2$. The expected number of k-vertex dense subgraphs formed by noise edges is

$$N = \binom{n}{k} \cdot \sum_{l=\lceil \delta s \rceil}^{s} \binom{s}{l} p^l (1-p)^{s-l}. \quad (2)$$

number of false patterns

When $n = 9{,}000$, $\delta = 0.5$, $k = 11$, $d = 5$,

$$p = 2\% \rightarrow N >> 1.$$

Partition: Using a Subset of Networks

$$b(m, \theta, q) = \sum_{l=\lceil \theta m \rceil}^{m} \binom{m}{l} q^l (1-q)^{m-l}.$$

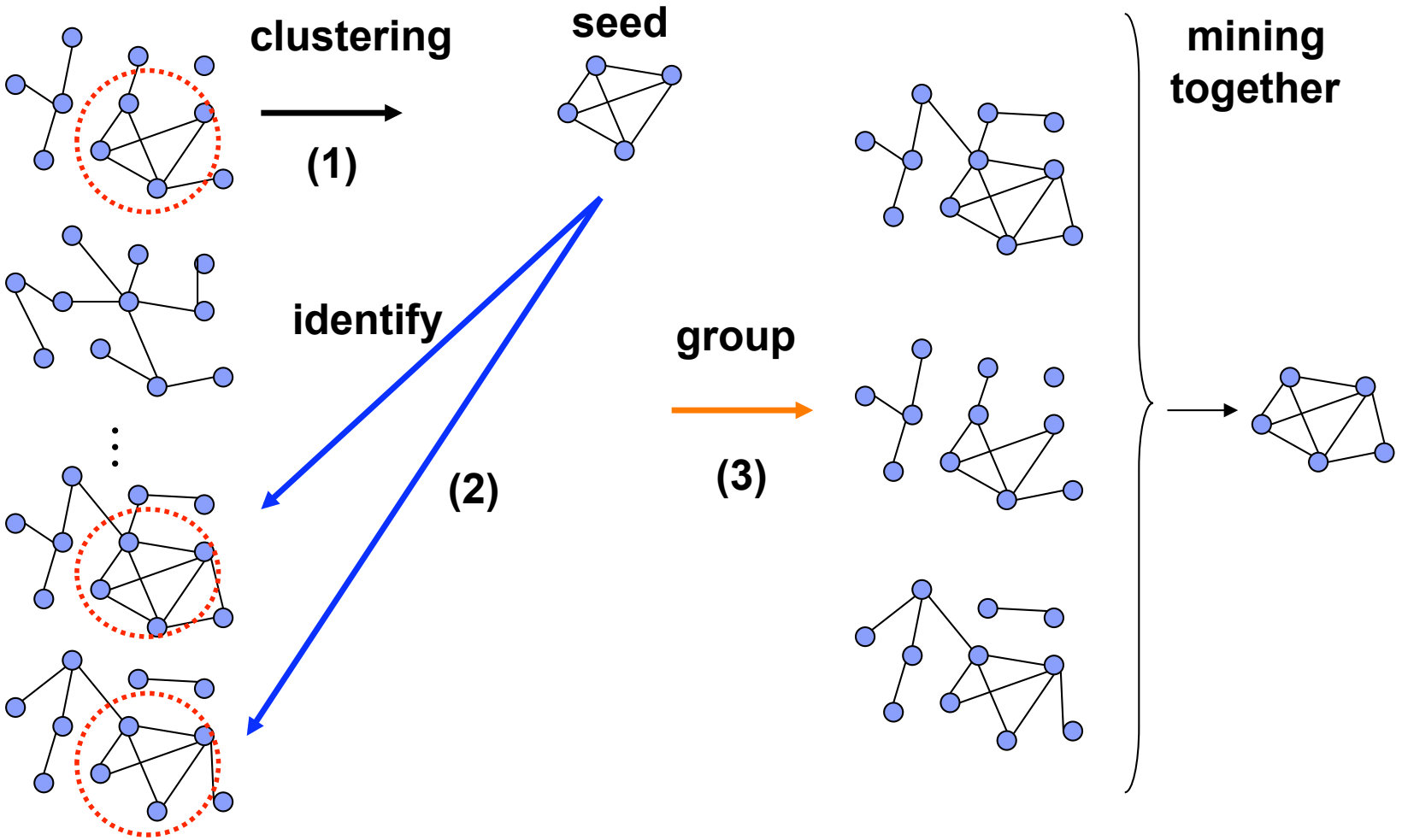
Use a subset of graphs if m ↓, then b ↓

→ Reduce the noise edge ratio (b) in summary graph

→ Reduce the number of false patterns

- How to choose a subset of networks? randomly select?
  100 choose 5 ≈ 75,287,520 subsets
- Unsupervised partition
- Supervised partition

Unsupervised Partition: Find a Subset

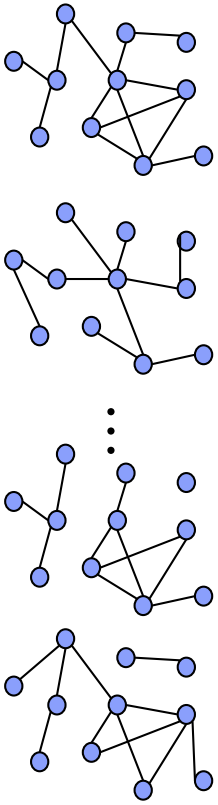


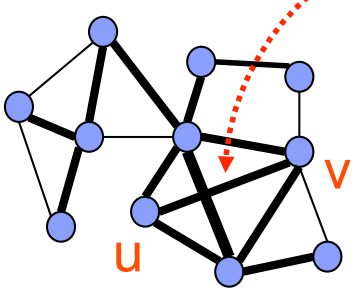
Neighbor Association: Change the Structure of Summary Graph

$$N = \binom{n}{k} \cdot \sum_{l=\lceil \delta s \rceil}^{s} \binom{s}{l} p^l (1-p)^{s-l}.$$

- Change the structure of summary graph, if p ↓, then N ↓
- Summary graph measures the association of vertices. In traditional summary graph, edge weight is determined by the number of edges that two vertices have in individual graphs.
- More stringent definition: the number of small frequent dense vertexsets (vertexlets)that two vertices belong to,
  → neighbor association summary graph

Neighbor Association Summary Graph





$$score(u,v) = \frac{\pi_{u,v}}{\pi_u}$$

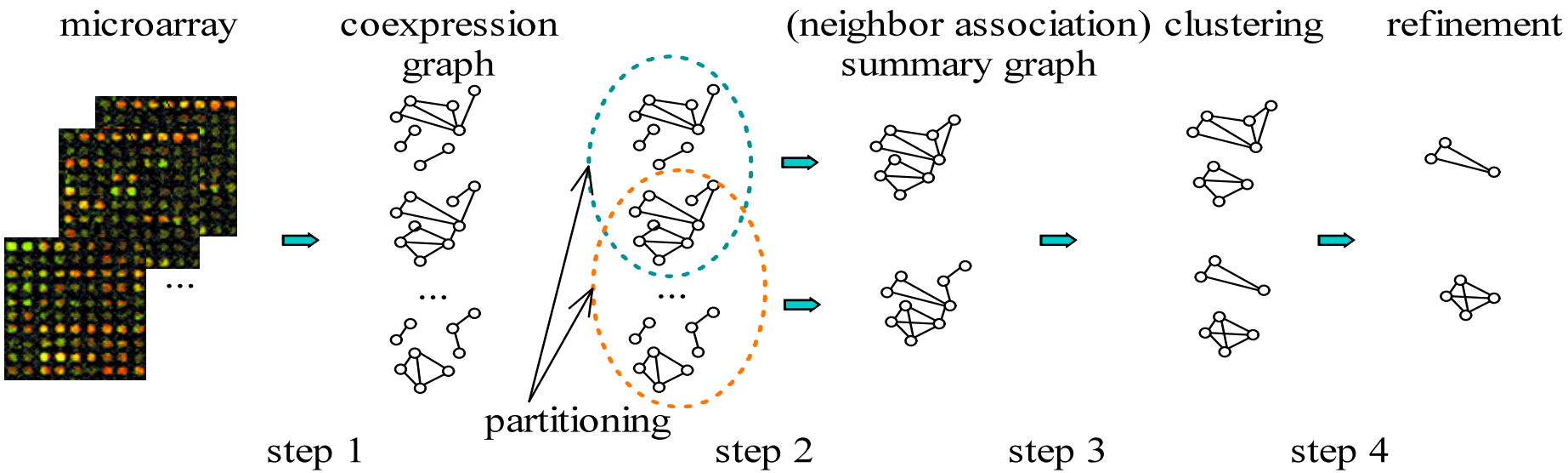
$\pi_{u,v}$: # of **frequent** dense vertexlets with k-1 nodes including *u* and *v*

π_u : # of frequent dense vertexlets with k nodes including *u*

$\pi_{u,v}$ is larger, u and v are more likely from the same module

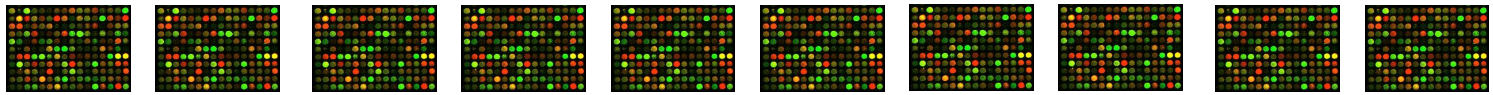
π_u normalization

The Complete Pipeline



Transcriptional Module Discovery

105 human microarray data sets



NeMo

4727 recurrent coexpression clusters
(density > 0.7 and support > 10)

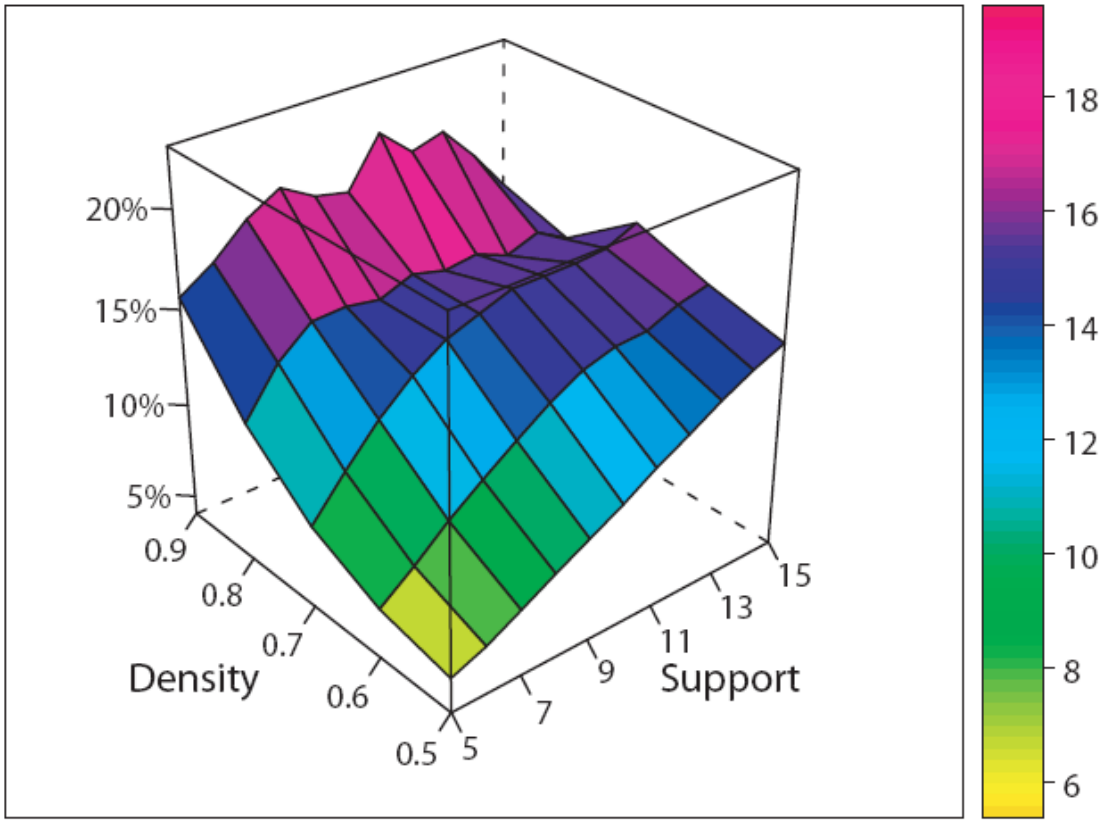
Validation based on ChIp-chip data
(9521 target genes for 20 TFs)

15.4% homogenous clusters
(vs. 0.2% by randomization test)

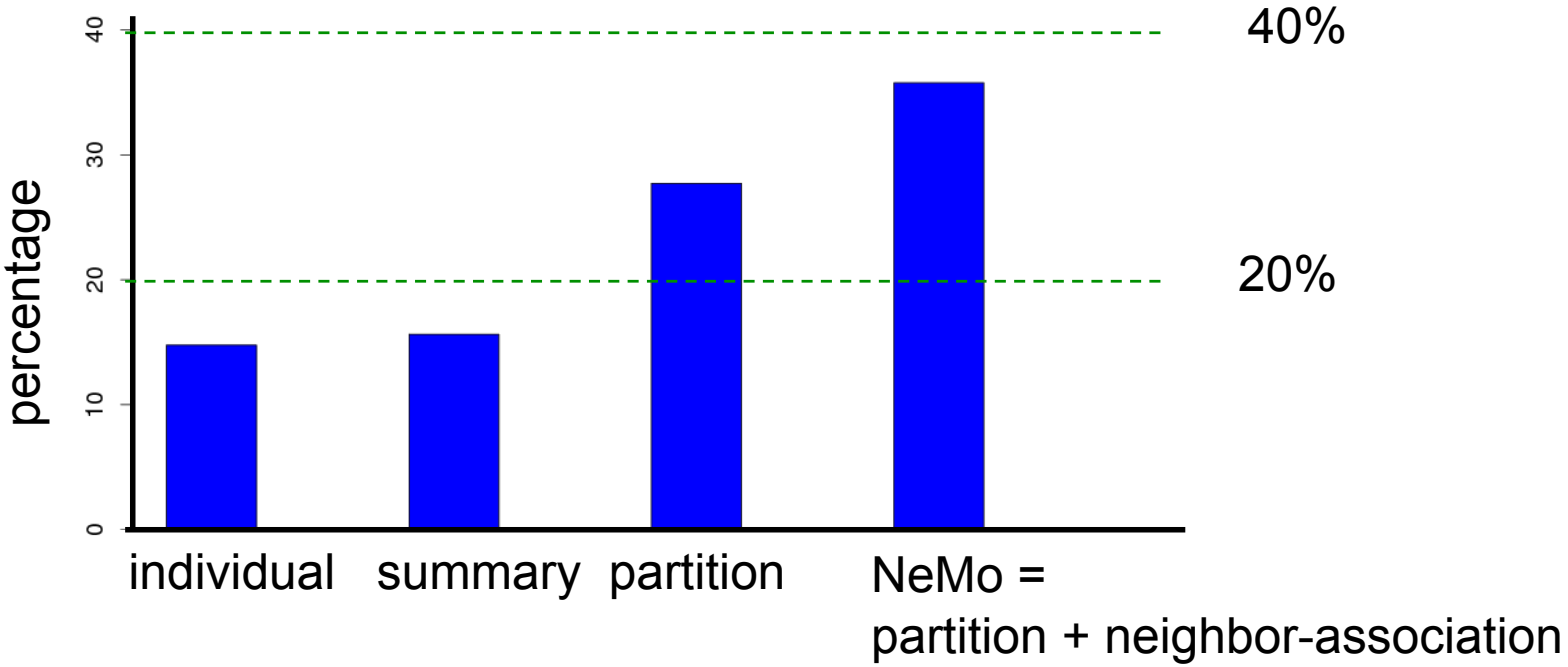
Validation based on human-mouse
Conserved Transfac prediction
(7720 target genes for 407 TFs)

12.5% homogenous clusters
(vs. 3.3% by randomization test)

Percentage of potential transcription modules validated by ChIP-Chip data increases with cluster density and recurrence



Performance Comparison



- individual < multiple
- partition works
- NeMo is better!

Conclusions

- Microarray data integration is important
  - Overcome the noise issue
- Microarray data integration is hard
  - Have the scalability issue
- NeMo: a graph-based approach
  - Partitioning
  - Neighbor Association Summary Graph