Finding Transcription Modules on Microarray Data Using PISA

Yangqiu Hu

12/15/2004

Outline

- Limitations of clustering
- Overview of biclustering
- Signature Algorithm (SA) and extensions (ISA, PISA)
- Implementation and results
- Conclusions

Microarray Data Analysis

- Classical clustering algorithms have been successful
  - Grouping genes of similar expression patterns
  - Global partitioning of the data
  - Generally a starting point in the analyses
  - E.g., hierarchical, *k*-means, SOM, …

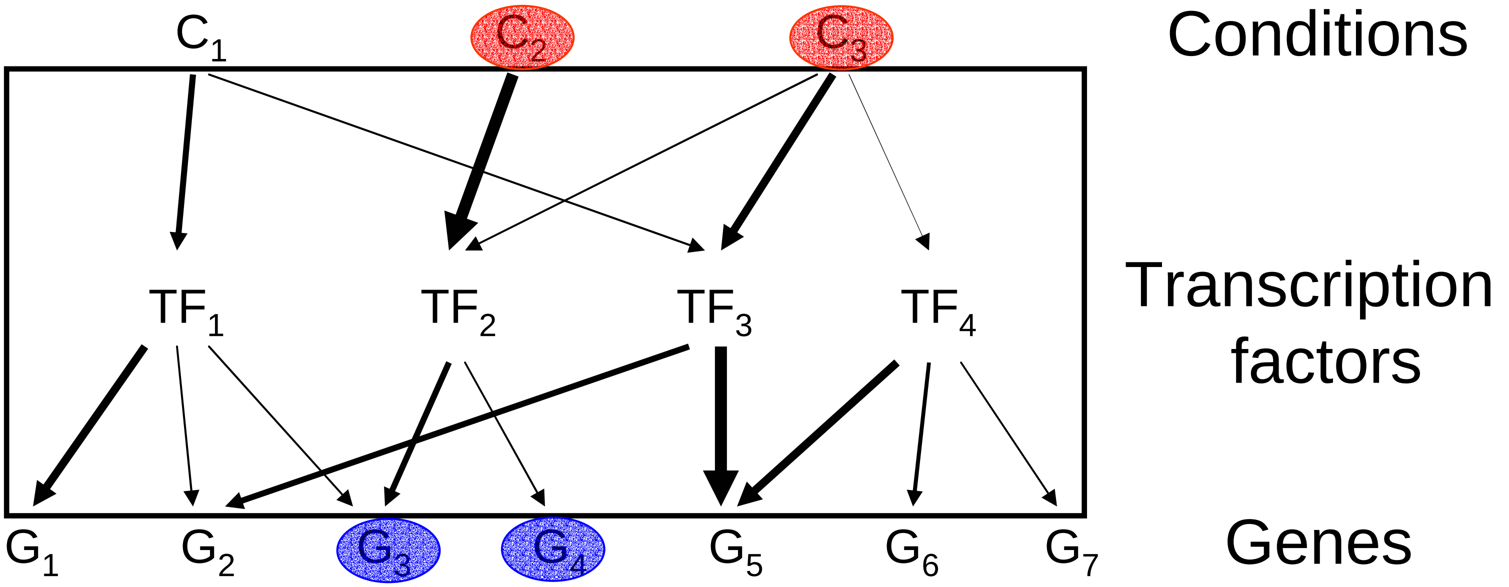
Limitations of Clustering

- Assigning each gene to a single cluster, while in fact many genes participate in several biological functions
- Measuring correlation over all conditions, but typically genes are only regulated in specific experimental context. Expression levels in uncorrelated conditions are simply *noise* for clustering

Biclustering

- Clustering both genes and conditions
- Overlapping clusters (vs. *disjoint* clusters)
- Local partitioning (vs. *global* partitioning)
- Other names:
  - Coclustering
  - Bidimensional clustering
  - Subspace clustering
  - Etc.

Transcription Modules



TM: a set of conditions and a set of genes connected by a transcription factor.

(From: Wingreen et al.)

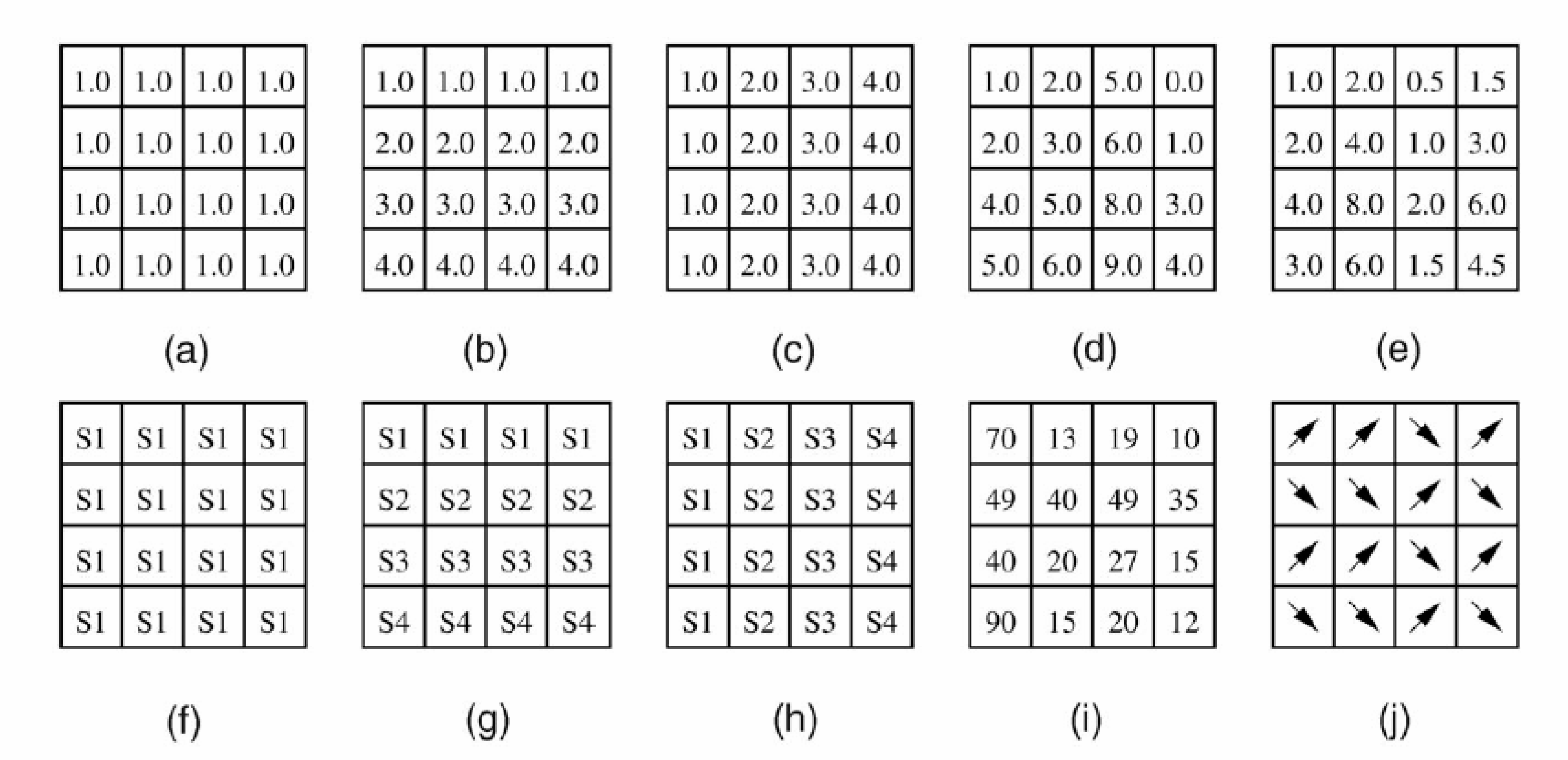
Finding Transcription Modules

- Transcription modules are
  - *Local* structures in microarray data matrix
  - *Non-exclusive*: they can overlap
  - *Non-exhaustive*: they do not have to cover all genes/conditions
- Classical clustering methods may have difficulties
- Biclustering methods may be used to find TM's

Overview of Biclustering

- Bicluster: a subset of rows that exhibit similar behavior across a subset of columns, and vice versa
- Biclustering: Given a data matrix, the identification of a set of biclusters that meet some homogeneity criteria
- Connection with weighted bipartite graph
- NP-complete – heuristic approaches

Bicluster Type



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

(a)

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

(b)

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

(c)

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

(d)

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

(e)

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

(f)

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

(g)

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

(h)

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

(i)

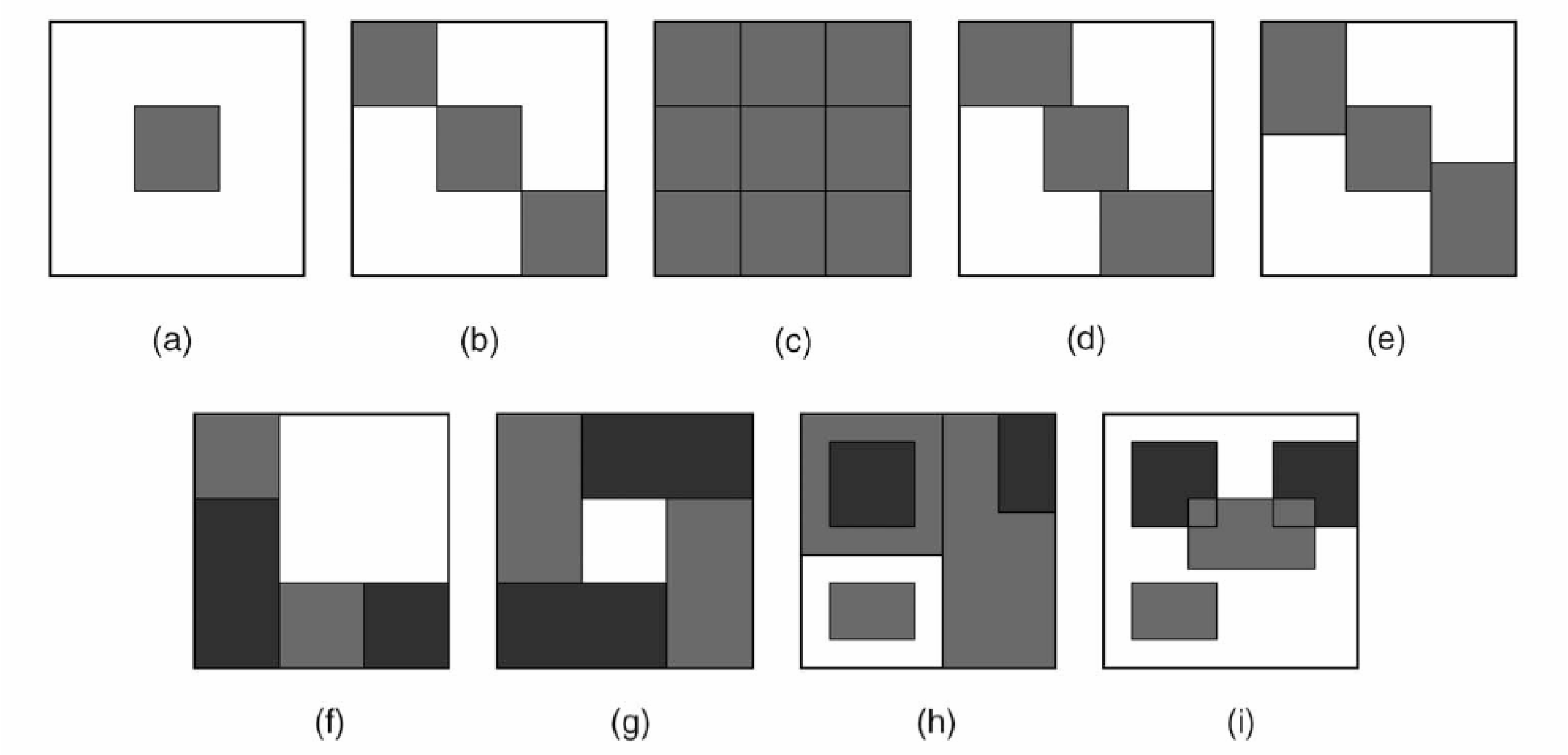
↗	↗	↘	↗
↘	↘	↗	↘
↗	↗	↘	↗
↘	↘	↗	↘

(j)

(a) Constant bicluster; (b) Constant rows; (c) Constant columns; (d) Coherent value (addictive); (e) Coherent value (multiplicative) (f) Overall coherent evolution; (g) Coherent evolution on rows; (h) Coherent evolution on columns; (i) Coherent evolution on columns (order preserving); (j) Coherent sign changes

(From: Madeira et al.)

Bicluster Structure



(a) Single bicluster; (b) Exclusive row/column; (c) Checkerboard; (d) Exclusive rows; (e) Exclusive columns (f) Non-overlapping with hierarchy; (g) Non-overlapping non-exclusive; (h) Overlapping with hierarchy; (i) Arbitrarily positioned overlapping

(From: Madeira et al.)

Some Biclustering Methods

- Cheng and Church
  - Coherent value, arbitrary overlapping
  - Greedy optimization of bicluster homogeneity
  - URL: http://cheng.ececs.uc.edu/biclustering/
- CTWC (Coupled Two-Way Clustering)
  - Coherent value, arbitrary overlapping
  - Separate row and column clustering
  - URL: http://ctwc.weizmann.ac.il/

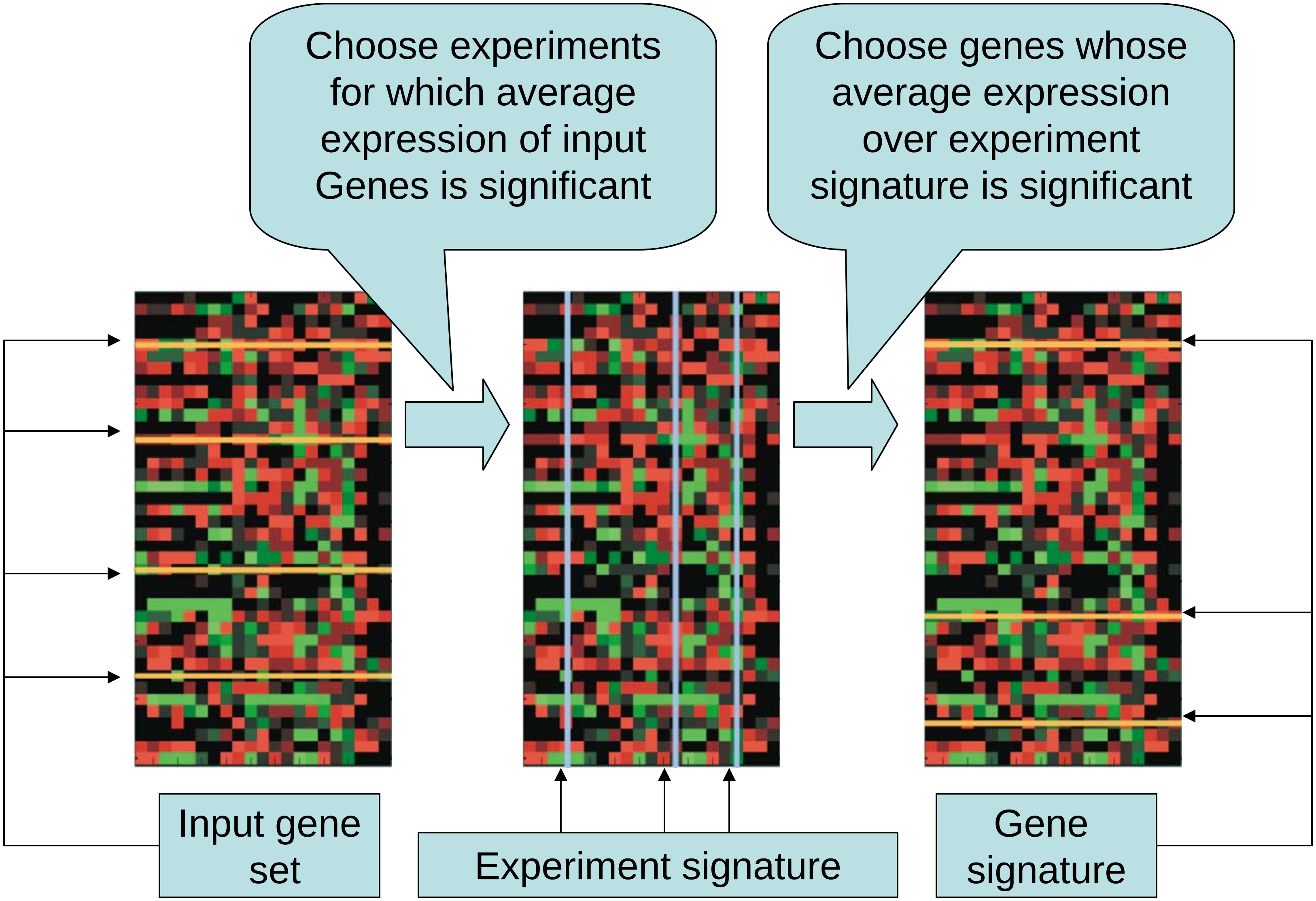
Some Biclustering Methods

- Plaid model
  - Coherent value, arbitrary overlapping
  - Distribution parameter estimation
  - URL: http://www-stat.stanford.edu/~owen/plaid/
- SAMBA
  - Coherent evolution, arbitrary overlapping
  - Bipartite graph
  - URL: http://www.cs.tau.ac.il/~rshamir/samba/

Signature Algorithm

- TM: a set of co-regulated genes and a set of conditions that trigger this co-regulation (Ihmels et al. 2002)
- Input: a set of genes that partially overlap a TM (prior information required)
- Output: a complete TM (gene signature + condition signature)

Signature Algorithm



Choose experiments for which average expression of input Genes is significant

Choose genes whose average expression over experiment signature is significant

Input gene set

Experiment signature

Gene signature

(From: Ihmels et al.)

Signature Algorithm

- Step 1: select the conditions under which the input genes are most tightly co-regulated

  – Condition score: $s_c = \left\langle E_G^{gc} \right\rangle_{g \in G_I}$

  – Thresholding:

$$S_C = \{ c \in C : \left| s_c - \left\langle s_c \right\rangle_{c \in C} \right| > t_C \sigma_C \}$$

Signature Algorithm

- Step 2: select the genes whose expression level change significantly from the whole genome under the conditions selected in step 1
  - Gene score: $s_g = \left\langle s_c E_C^{gc} \right\rangle_{c \in S_c}$
  - Thresholding:

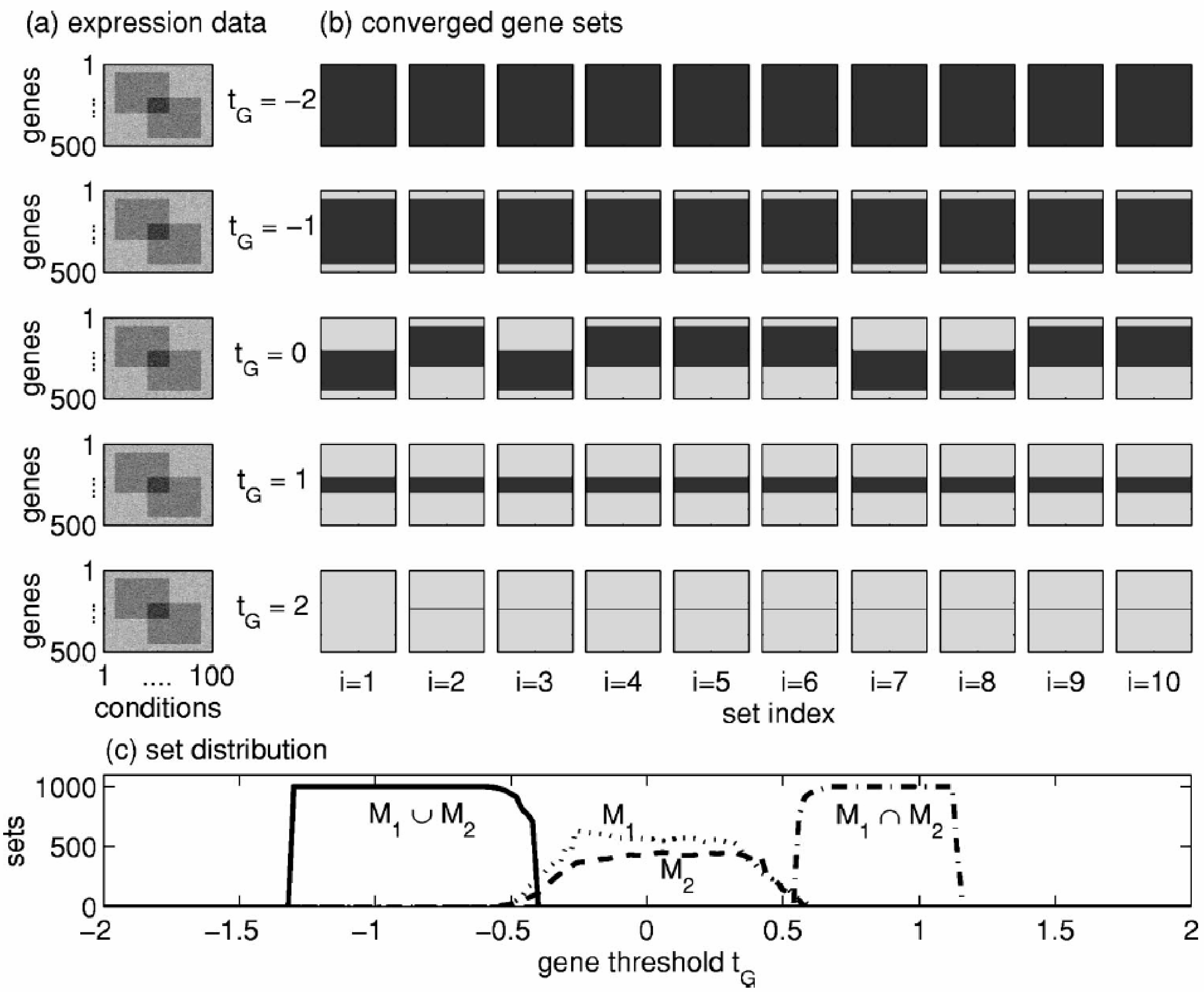
$$S_G = \{g \in G : \left| s_g - \left\langle s_g \right\rangle_{g \in G} \right| > t_G \sigma_G \}$$

Signature Algorithm

- Symmetric in genes and conditions
- Uncorrelated genes/conditions will be removed
- Disadvantages:
    - Requires prior knowledge
    - How to choose the threshold values
    - Only two steps: no further iteration

Iterative Signature Algorithm

- ISA extends SA by
  - Running SA iteratively
  - Starting with random input gene sets
  - Using a range of threshold values
- Advantages of ISA:
  - Requires no prior knowledge
  - Reveals the hierarchical modular organization at different resolutions



(From: Bergman et al.)

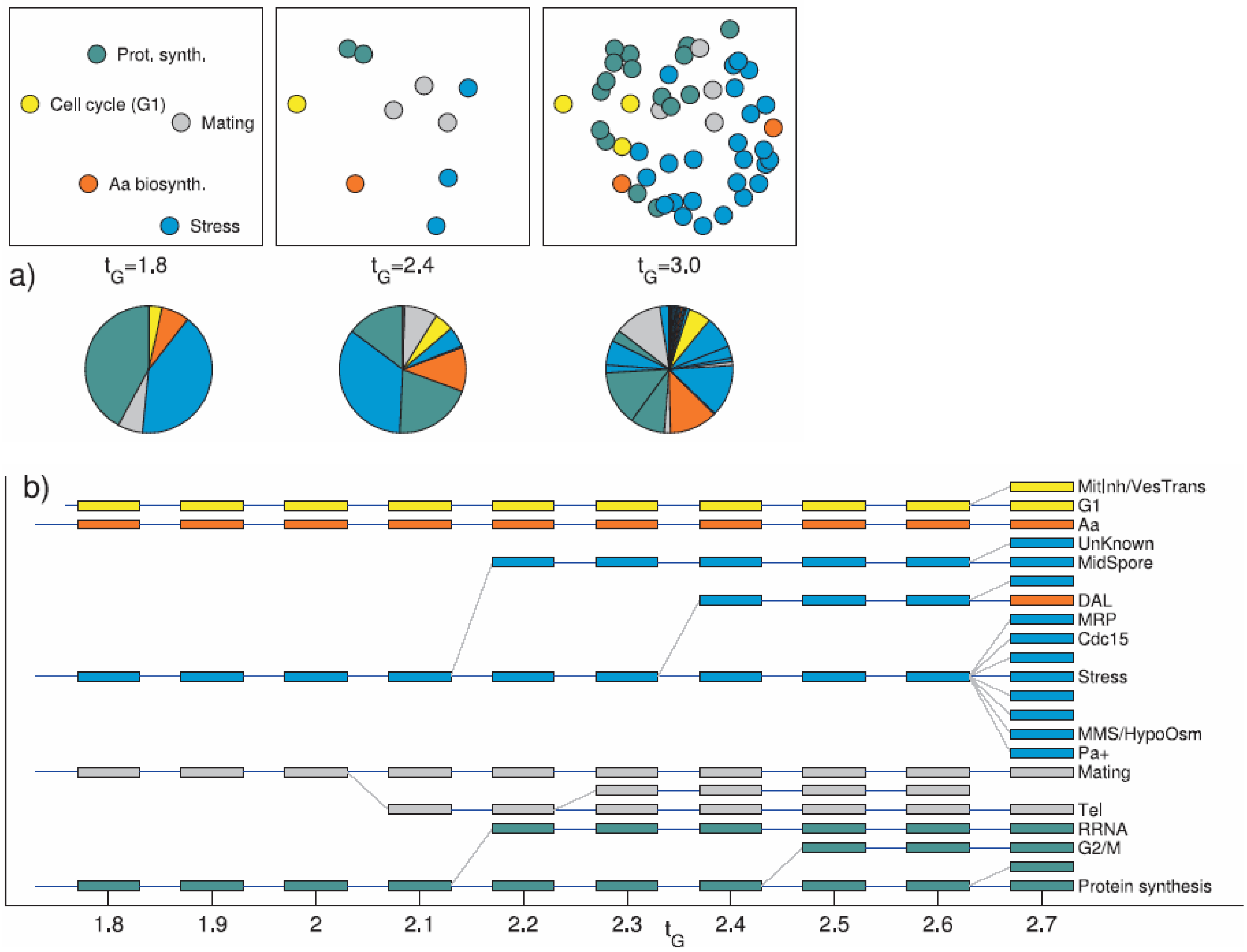
ISA Applied to Yeast Data

- *Saccharomyces Cerevisiae* microarray data containing 6206 genes and 1011 experimental entries
- Using t_G = 1.8, 1.9, …, 4.0, and t_C = 2.0
- Using ~20,000 random input gene sets, each generating a fixed point per t_G
- Module fusion: agglomerative clustering of the fixed points for each t_G

ISA Results

- 2956 out of 6206 genes are included in at least one module, with a few overlapping
- All experimental conditions are associated with at least one module, with large overlapping
- Module size is between 100~300 genes
- t_G ↑ , module size ↓ , # of modules ↑ (higher resolution)

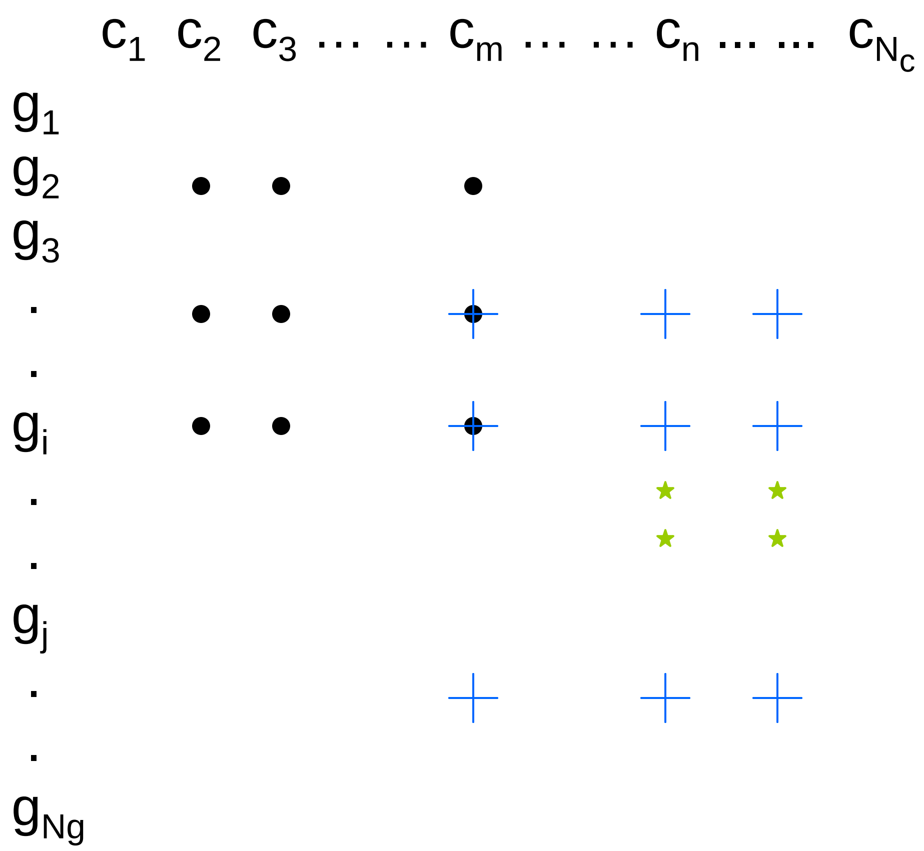
Hierarchical Modular Organization



(From: Ihmels et al.)

Limitations of ISA

- Lots of spurious modules
- Weak modules may be *overwhelmed* by strong modules



(From: Wingreen et al.)

Progressive Iterative Signature Algorithm (PISA)

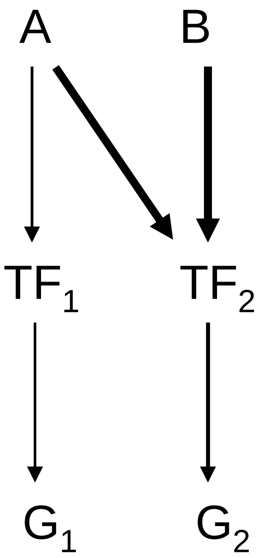
- Removes the contributions of the already found module to the expression data
- Reduces positive feedback due to random input sets
- Improves thresholding on gene scores, no thresholding on condition scores

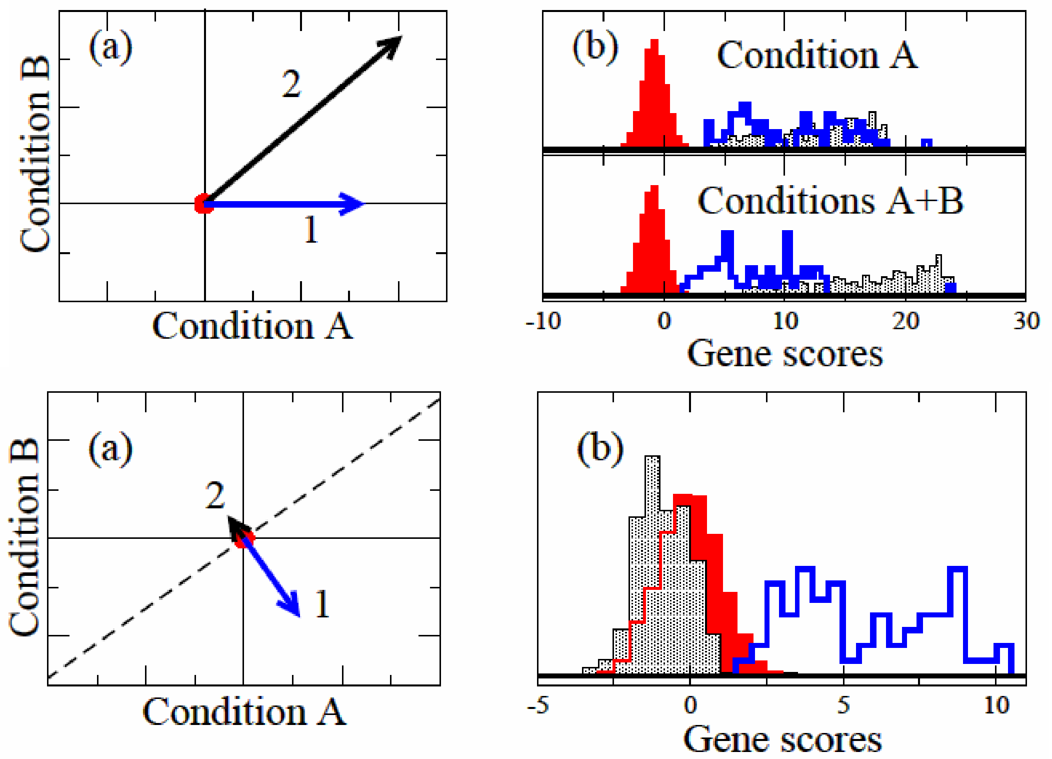
PISA Implementation

- Normalization of expression data
  - Making gene scores comparable for thresholding (E → E_G and E_C)
- PISAstep
  - Modified ISA
- ➢ Orthogonalization:
  - Removing found module
- ➢ Postprocessing:
  - Preliminary modules → consistent modules

Orthogonalization (1)

- Each condition score vector S^C is required to be orthogonal to that of the previously found modules





(From: Kloster et al.)

Orthogonalization (2)

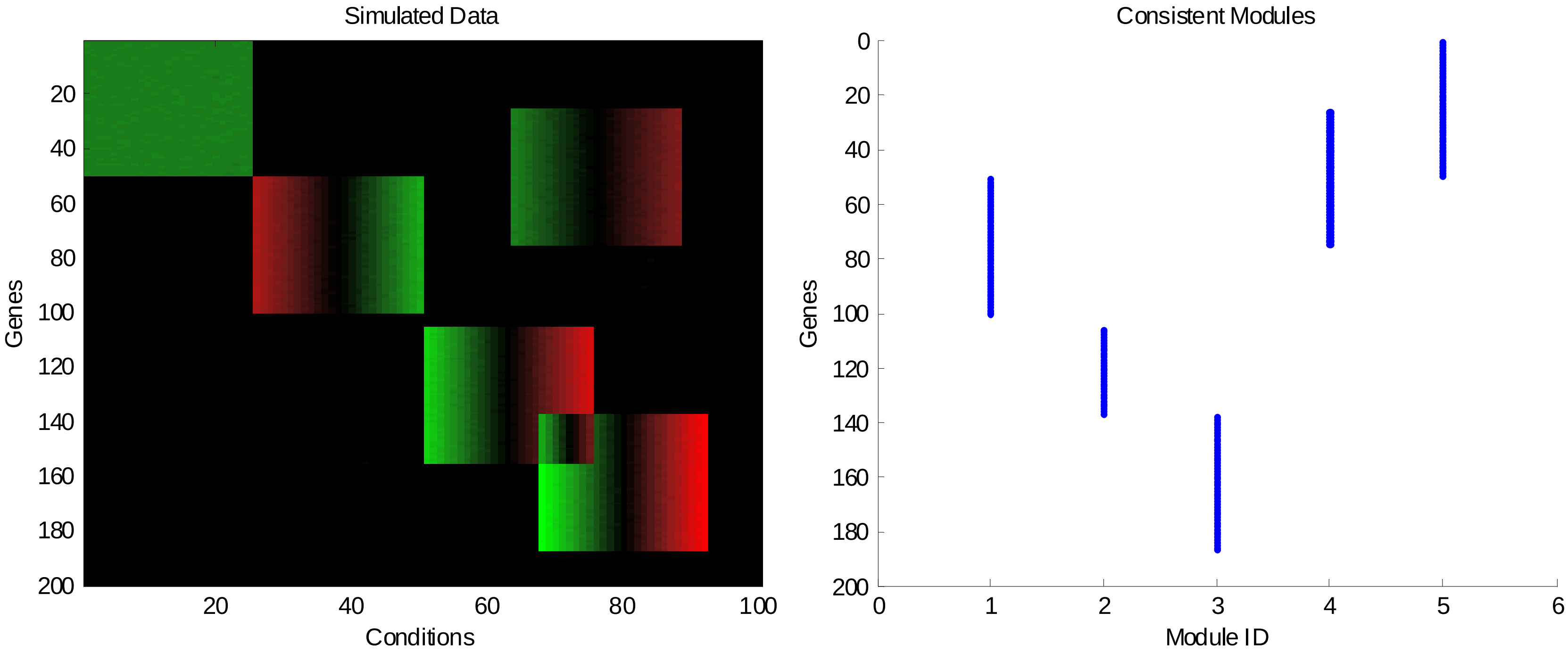
- After finding a module (S^G, S^C), remove the component along S^C for all genes:

$$E_C^{new} = E_C - E_C \frac{S^C \left(S^C\right)^T}{\left|S^C\right|^2}$$

Finding Consistent Modules

- Run PISA many times (~100)
- Tabulate all preliminary modules (fixed points)
- Consistency check:
  - PM has > 50% genes in the CM
  - Genes appear in > 20% of the PMs
  - Iterate …
- Our approach:
  - Clustering the condition scores of PMs

Results – Simulated Data



One of the overlapping modules, module #2, is incomplete

Results – Yeast Expression Data

- Expression data from Gasch et. al., *Genomic expression programs in the response of yeast cells to environmental changes*, Mol Biol Cell. 2000 Dec;11(12):4241-57, with 6152 genes and 173 conditions

- For comparison, only use those genes as in Segal et. al. *Module Networks: Identifying Regulatory Modules and their Condition Specific Regulators from Gene Expression Data*, Nat Genet. 2003 Jun;34(2):166-76, with 2355 genes and 173 conditions

- Segal et. al. identified 50 non-overlapping modules using their PCluster (Probabilistic Agglomerative Clustering)

Results – Yeast Expression Data

- We ran PISA 100 times and got 2210 preliminary modules
- Our postprocessing method allows to determine the # of consistent modules
- 30 minutes on PC, Matlab implementation

# mod.	% genes included	max # overlapping mod.	mean mod. size
50	78.28%	11	99.76
100	89.20%	16	91.72
150	94.60%	24	95.31

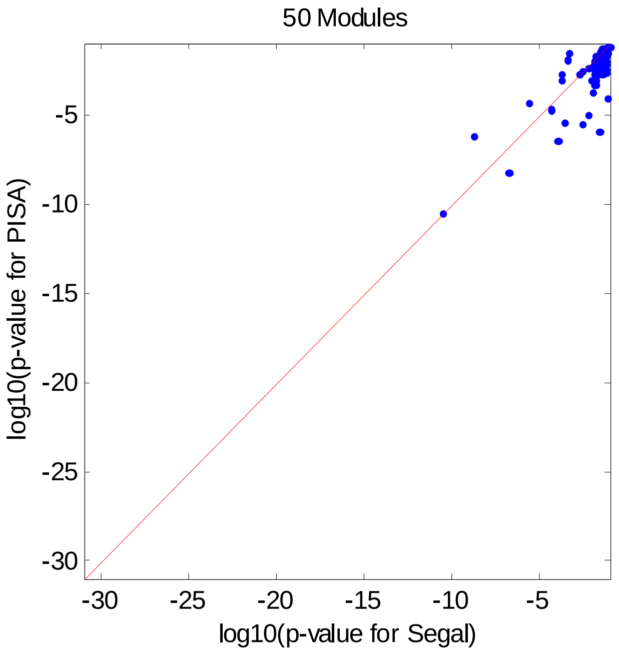
Performance Comparison

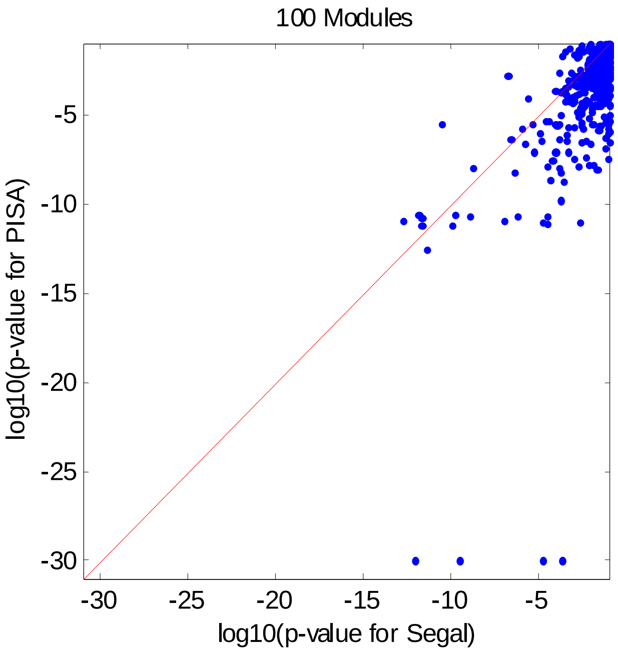
- Biological relevance using Gene Ontology

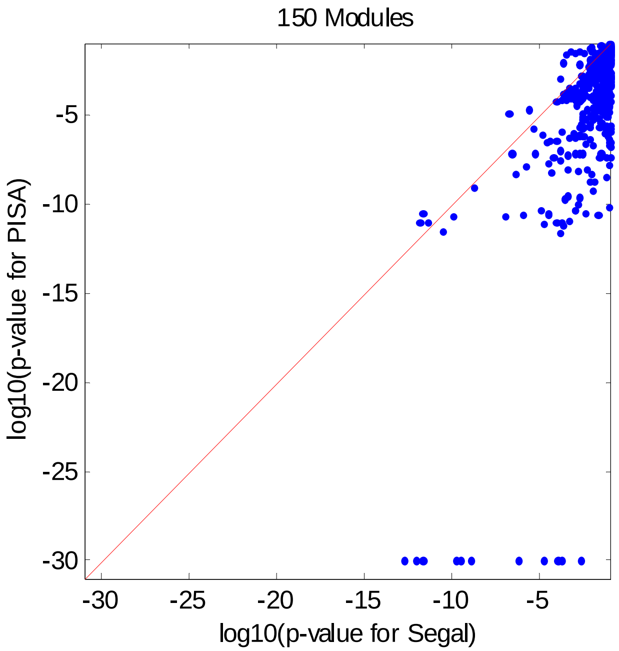
$$p = 1 - \sum_{i=0}^{n-1} \frac{\binom{c}{i}\binom{N_G - c}{m - i}}{\binom{N_G}{m}}$$

N_g — number of genes in organism (2355)
m — number of genes in module
c — number of genes in GO category
n — number of genes in both module and GO category

Performance Comparison







Only GO categories with no more than 300 genes are used for computing the p-values

Conclusions

- Classical clustering methods may encounter difficulties when applied to microarray data with large # of samples
- Would biclustering be a promising solution?
- Judging from the overlap with GO annotations, PISA’s results on the yeast expression data are better than those in the original paper

Future Work

- Determining the optimal # of modules
- Applying PISA to more data sets
- Validation of biclustering methods, using both internal and external data
- Comparing PISA with other biclustering methods
- …