Clustering gene expression data

How Gene Exp ession Data Lo

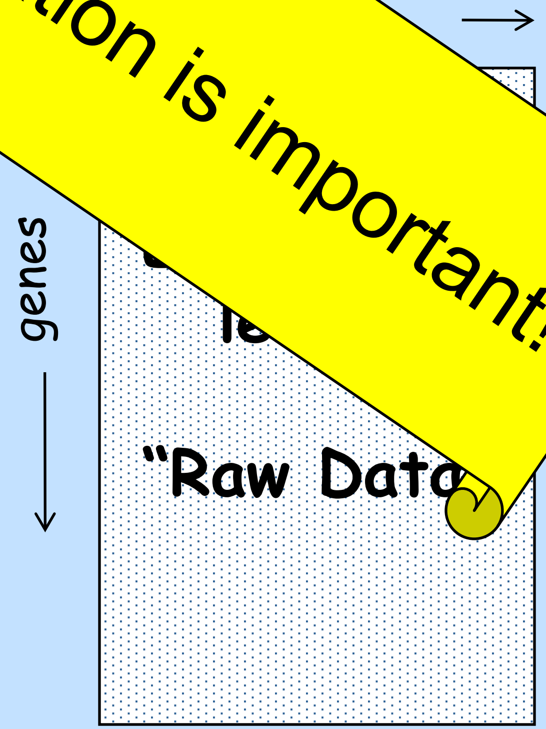
Normalization is important!!

Entries of the Raw Data m

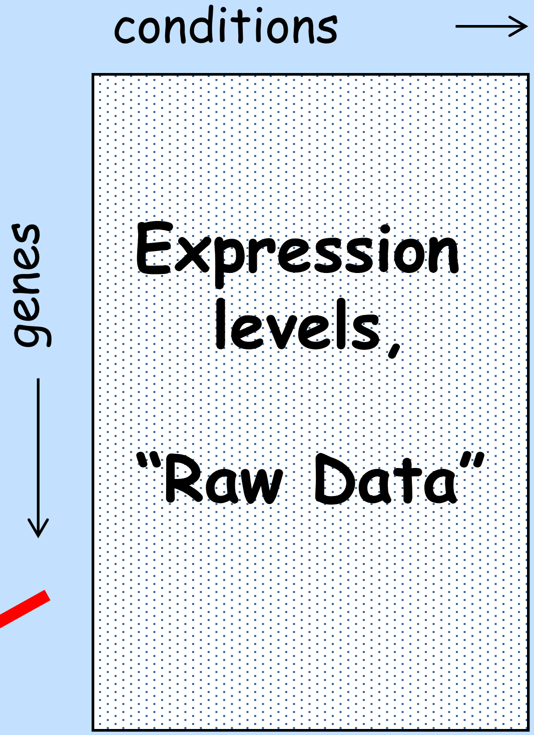
- Ratio values
- Absolute values
- …

- Row = gene's expression pattern

- Column = experiment/condition's profile



Data Preprocessing



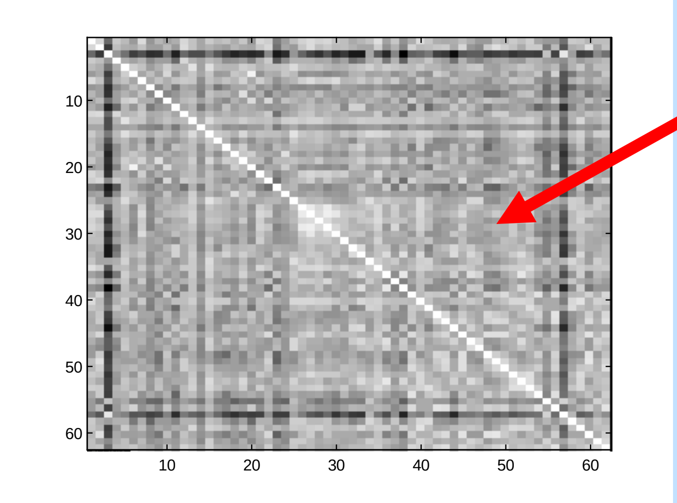
conditions →

genes ↓

Expression levels,

"Raw Data"

- **Input:** Real-valued raw data matrix.
- **Compute the similarity matrix** (cosine angle/correlation/…)
- Alternatively – distances



From the Raw Data matrix we compute the similarity matrix S.
S_{ij} reflects the similarity of the expression patterns of gene *i* and gene *j*.

DNA chips: Applications

- Deducing functions of unknown genes (similar expression pattern ⇨ similar function)
- Deciphering regulatory mechanisms (co-expression ⇨ co-regulation).
- Identifying disease profiles
- Drug development
- …

Analysis requires clustering of genes/conditions.

Clustering: Objective

Group elements (genes) to clusters satisfying:

- **Homogeneity**: Elements inside a cluster are highly similar to each other.

- **Separation**: Elements from different clusters have low similarity to each other.

- Unsupervised (no labels).
- Most formulations are NP-hard (e.g. minimum clique cover).

The Clustering Bazaar

Hierarchical clustering

An Alternative View

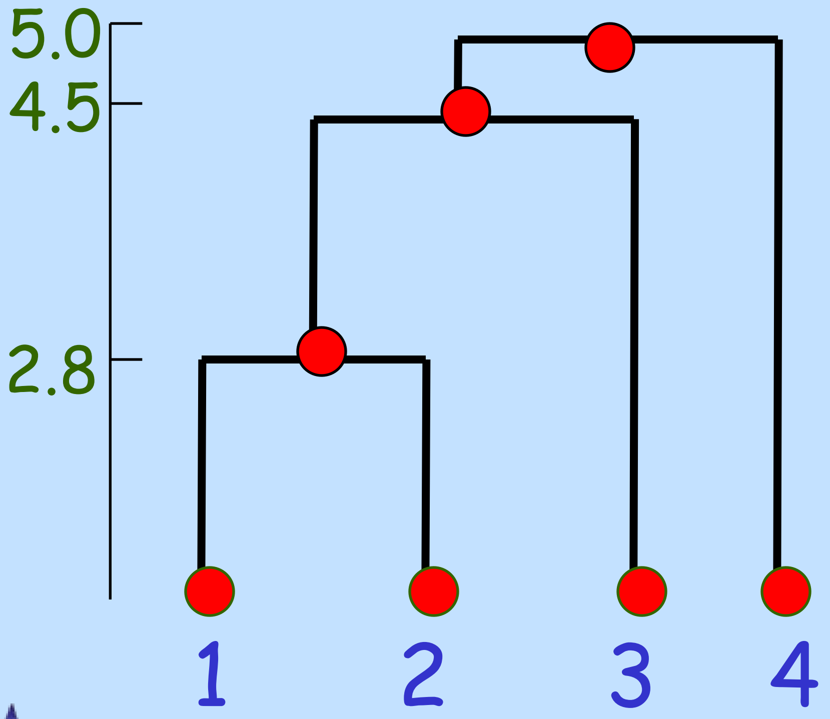
Instead of partition to clusters – Form a tree-hierarchy of the input elements satisfying:

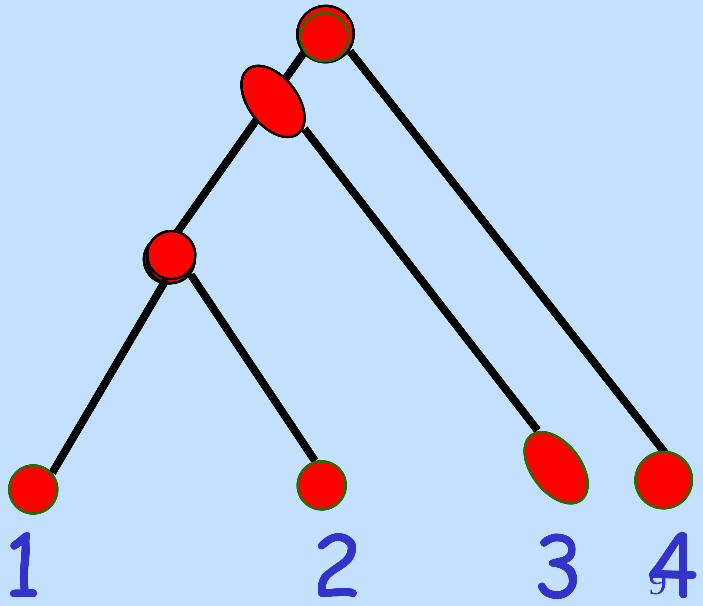
- More similar elements are placed closer along the tree.

- Or: Tree distances reflect distance between elements

Hierarchical Representation

Dendrogram: rooted tree, usually binary; all leaf-root distances are equal. Ordinates reflect (avg.) distances between the corresponding subtrees.





Hierarchical Clustering: Average Linkage
Sokal & Michener 58, Lance & Williams 67

- Input: Distance matrix (D_{ij})
- Iterative algorithm. Initially each element is a cluster. n_r- size of cluster r
  - – Find min element D_{rs} in D; merge clusters r,s
  - – Delete elements r,s; add new element t with $D_{it}=D_{ti}=n_r/(n_r+n_s)\bullet D_{ir}+ n_s/(n_r+n_s) \bullet D_{is}$
  - – Repeat

Average Linkage (cont.)

- Claim: D_{rs} is the average distance between elements in *r* and *s*.
- Proof by induction…
- Claim: D_{rs} can only increase.

A General Framework

Lance & Williams 67

- Find min element D_{rs} , merge clusters r,s
- Delete elems. r,s, add new elem. t with

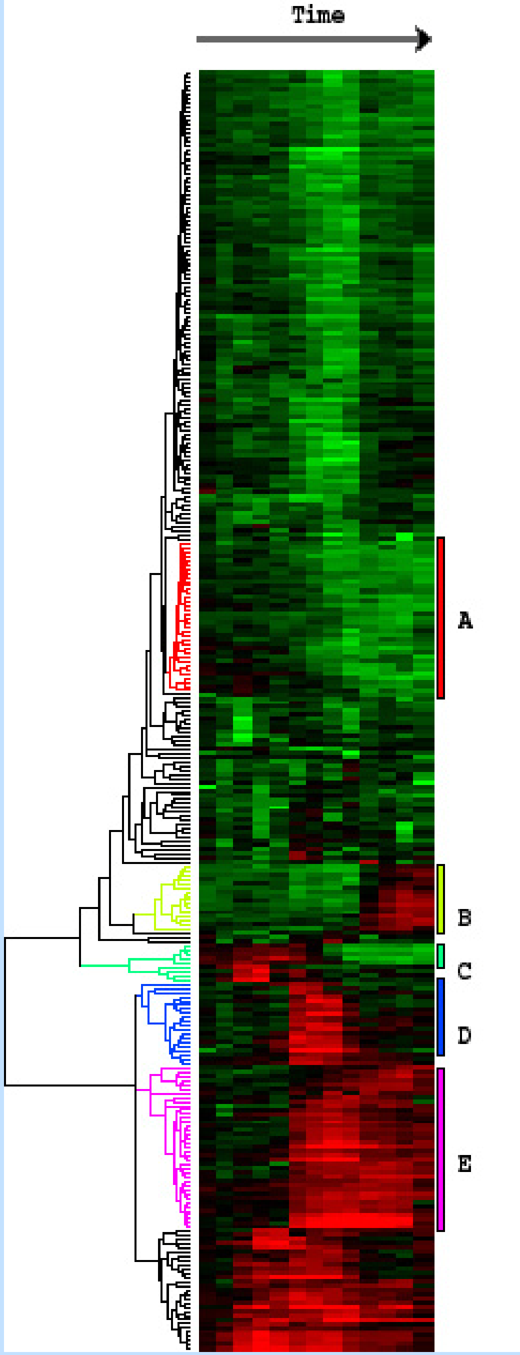
  $D_{it}=D_{ti}=\alpha_r D_{ir}+ \alpha_s D_{is} + \gamma|D_{ir}-D_{is}|$

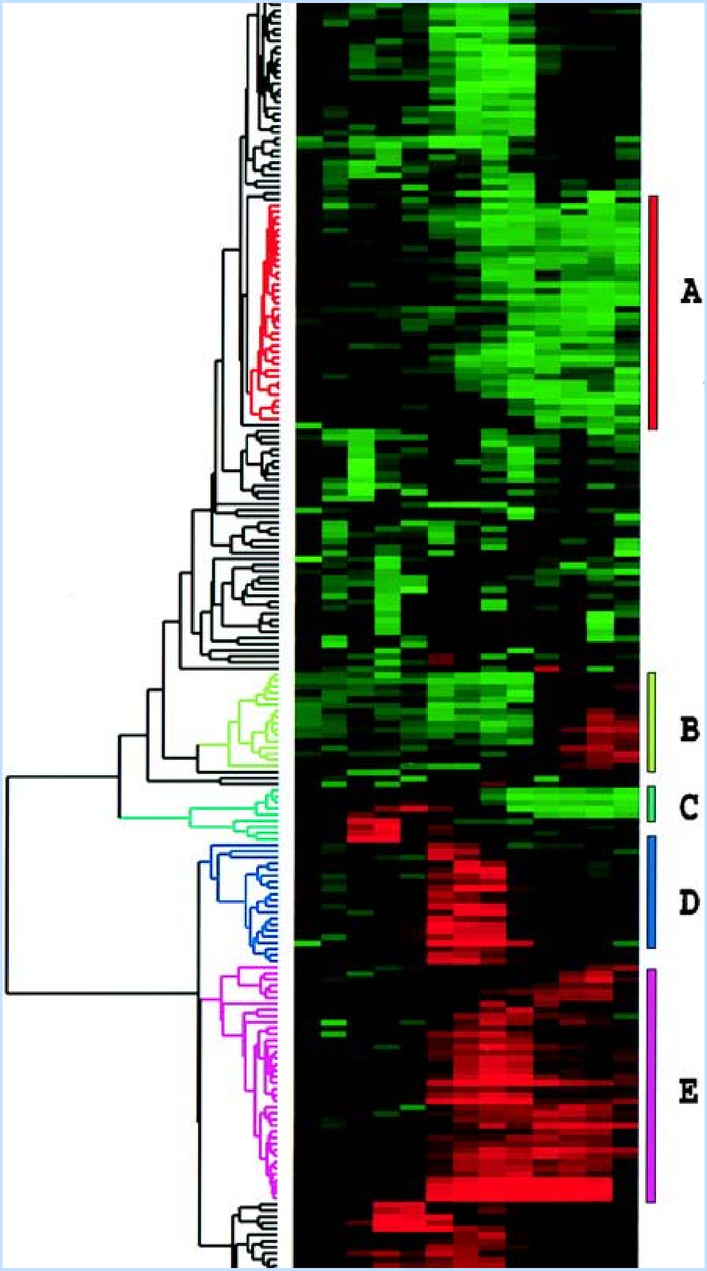
- Single-linkage: $D_{it}=\min\{D_{ir},D_{is}\}$
- Complete-linkage: $D_{it}=\max\{D_{ir},D_{is}\}$

Hierarchical clustering of GE data

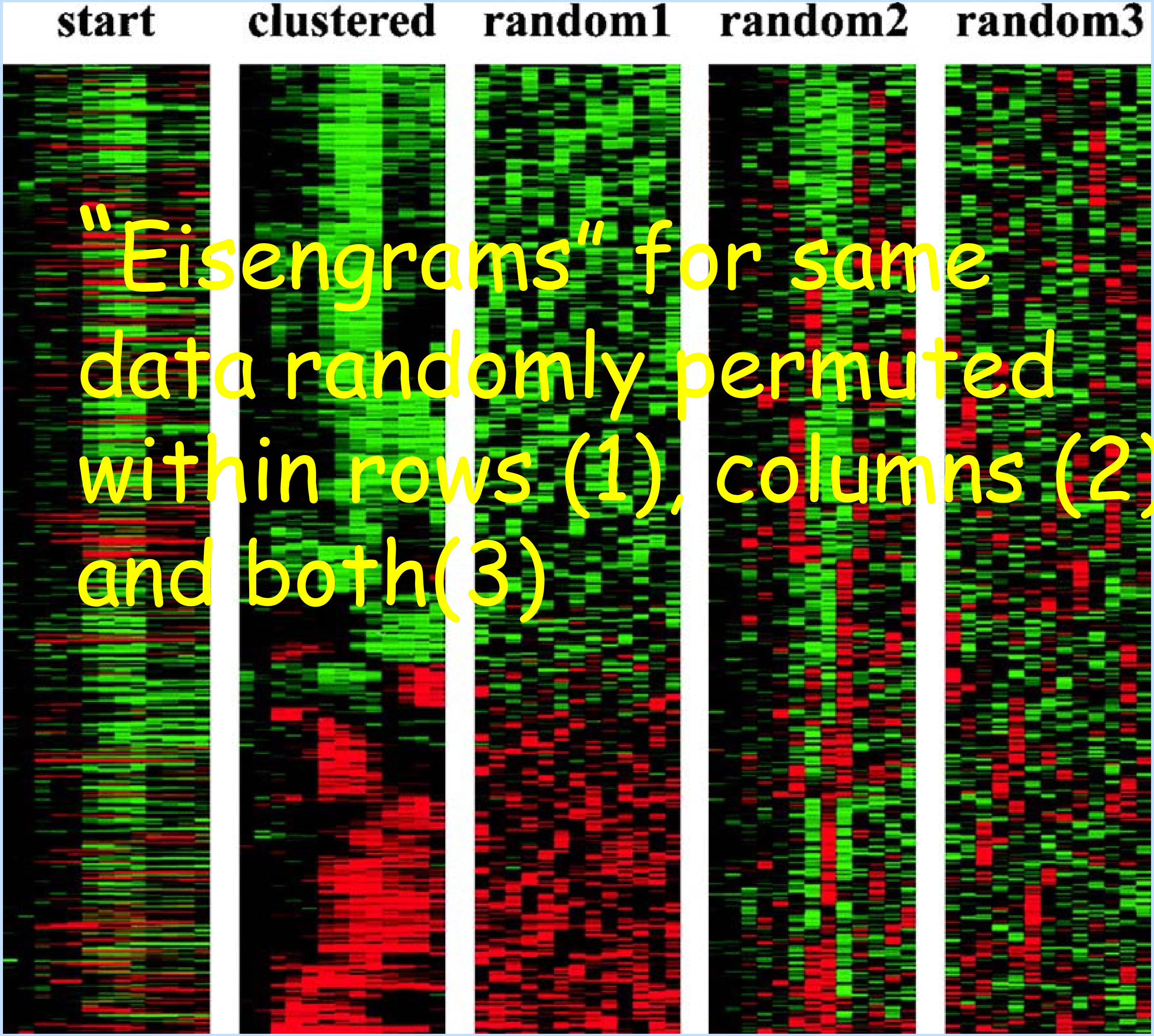
Eisen et al., PNAS 1998

- Growth response: Starved human fibroblast cells, added serum
- Monitored 8600 genes over 13 time-points
- t_{ij} - fluorescence level of gene i in condition j; r_{ij} – same for reference (time=0).
- $s_{ij} = \log(t_{ij}/r_{ij})$
- $S_{kl}=(\Sigma_j s_{kj} \bullet s_{lj})/[|s_k||s_l|]$ (cosine of angle)
- Applied average linkage method
- Ordered leaves by increasing average expression level (or other criteria)





CG



"Eisengrams" for same data randomly permuted within rows (1), columns (2) and both(3)

Comments

- Distinct measurements of same genes cluster together
- Genes of similar function cluster together
- Many cluster-function specific insights
- Interpretation is a REAL biological challenge

More on hierarchical methods

- Agglomerative vs. the “more natural” divisive.
- Advantages:
  - gives a single coherent global picture
  - Intuitive for biologists (from phylogeny)
- Disadvantages:
  - No single partition; no specific clusters
  - Forces all elements to fit a tree hierarchy

Non-Hierarchical Clustering

K-means (Lloyd' 57, Macqueen '67)

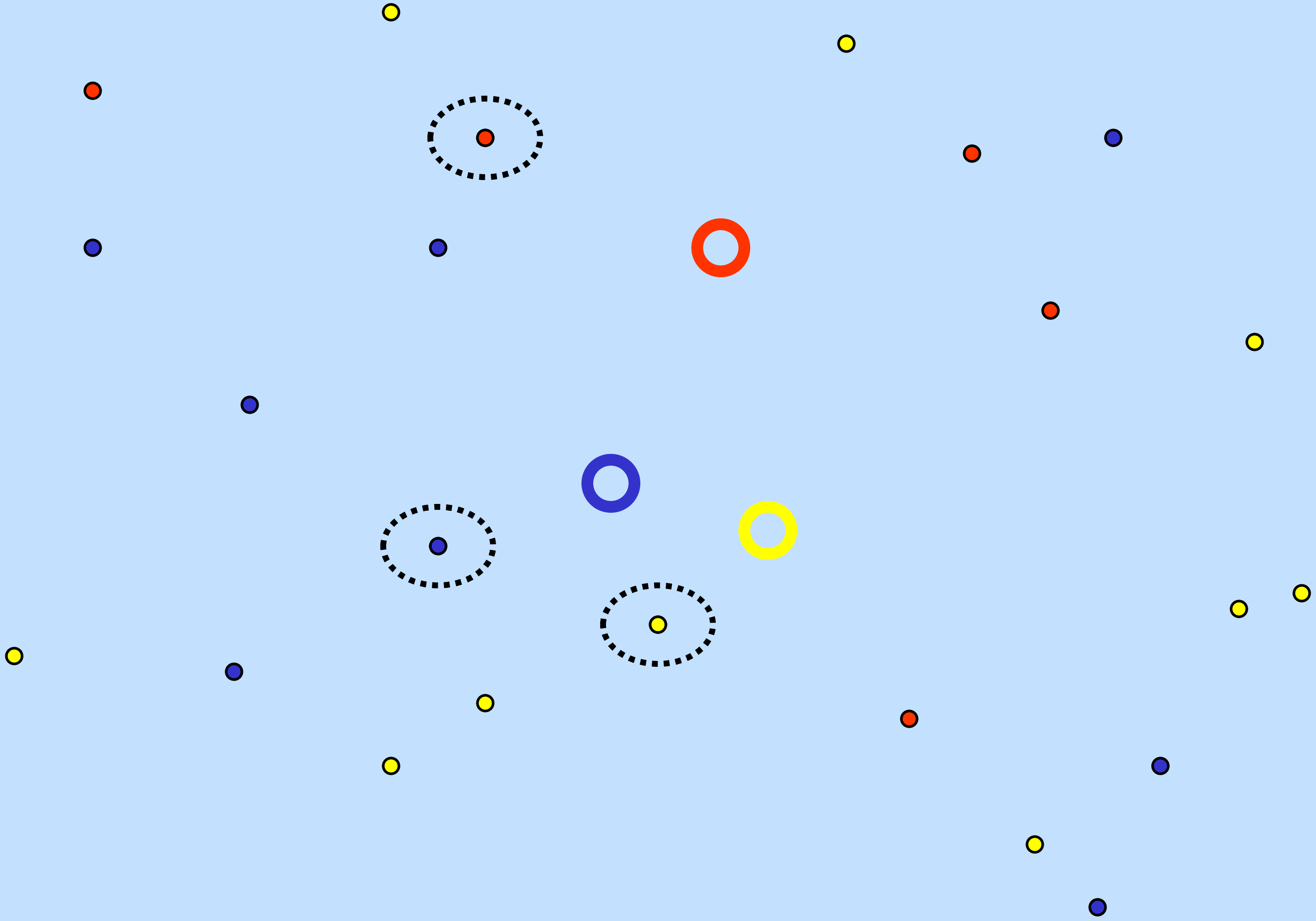
- Input: vector v_i for each element i; #clusters=k
- Define a centroid c_p of a cluster C_p as its average vector.
- Goal: minimize $\Sigma_{clusters\ p}\Sigma_{i\ in\ cluster\ p}\,d(v_i,c_p)$

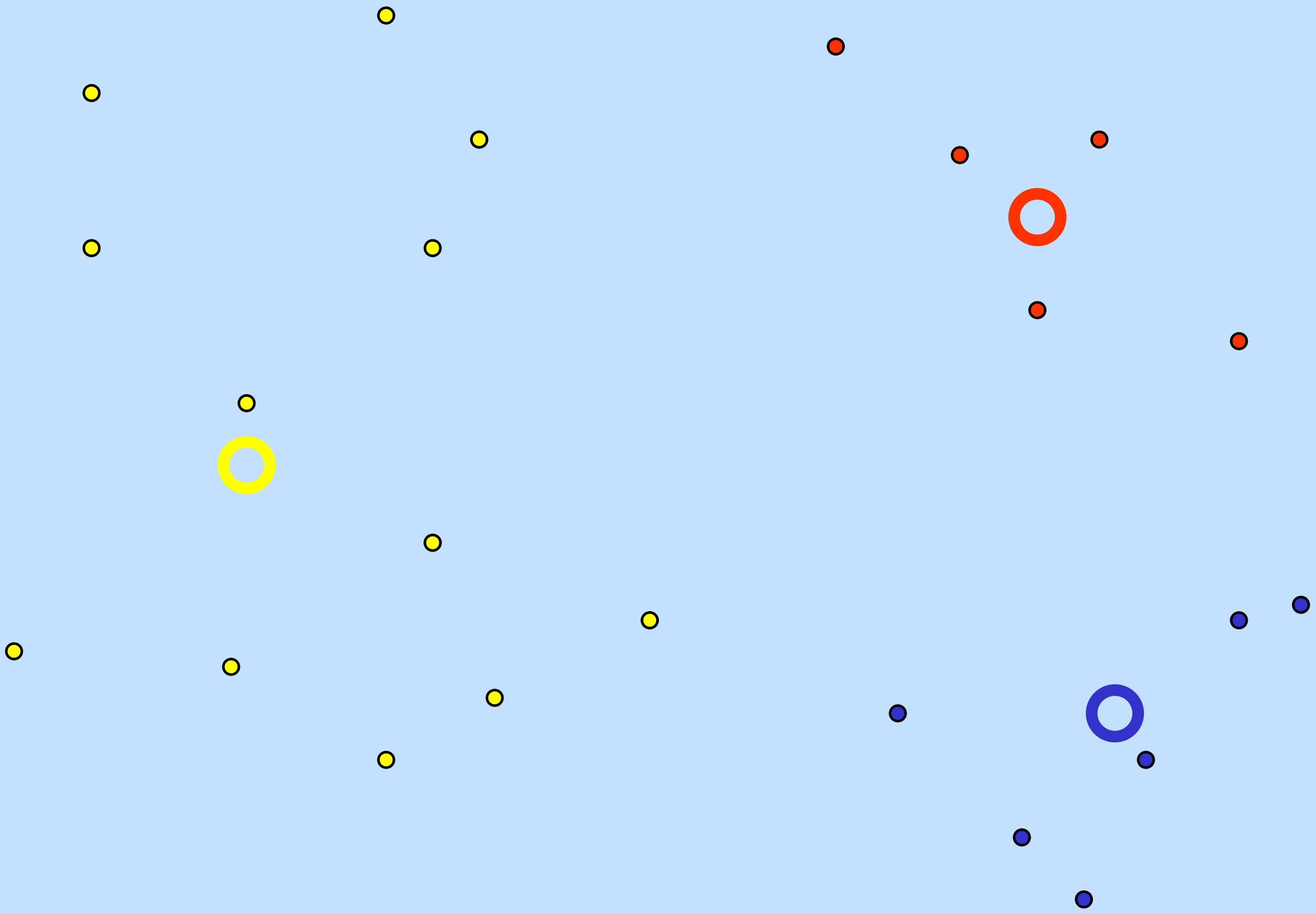
- Objective = homogeneity only (k fixed)
- NP-hard already for k=2.

K-means alg.

- Initialize an arbitrary partition P into k clusters.
- Repeat the following till convergence:
  - Update centroids (max c, P fixed)
  - Assign each point to its closest centroid (max P, c fixed)

- Can be shown to have poly expected time under various assumptions on data distribution.
- A variant: perform a single best modification (that decreases the score the most).





A Soft Version

- Based on a probabilistic model of data as coming from a mixture of Gaussians: $P(z_i = j) = \pi_j$

  $P(x_i \mid z_i = j) \sim N(\mu_j, \sigma^2 I)$
- Goal: evaluate the parameters θ (assume σ is known).
- Method: apply EM to maximize the likelihood of data.

$$L(\theta) \propto \prod_i \sum_j \pi_j \exp(-\frac{d(x_i, \mu_j)^2}{2\sigma^2})$$

K-means, soft version

- Iteratively, compute soft assignment and use it to derive expectations of π, μ:

$$w_{ij}^{(t)} = p(z_i = j|\mathbf{x}_i, \theta^{(t)}) = \frac{\pi_j^{(t)} p(\mathbf{x}_i|z_i = j, \theta^{(t)})}{\sum_{k=1}^{n} \pi_k^{(t)} p(\mathbf{x}_i|z_i = k, \theta^{(t)})}$$

$$\pi_j^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} w_{ij}^{(t)}$$

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} w_{ij}^{(t)} \mathbf{x}_i}{\sum_{i=1}^{N} w_{ij}^{(t)}}$$

Soft vs. hard k-means

Soft EM optimizes:

$$\Theta^* \quad = \quad \operatorname*{argmax}_{\Theta} \sum_{z_1,\ldots,z_n} P_\Theta(x_1,\ldots,x_n,z_1,\ldots,z_n)$$

Hard EM optimizes:

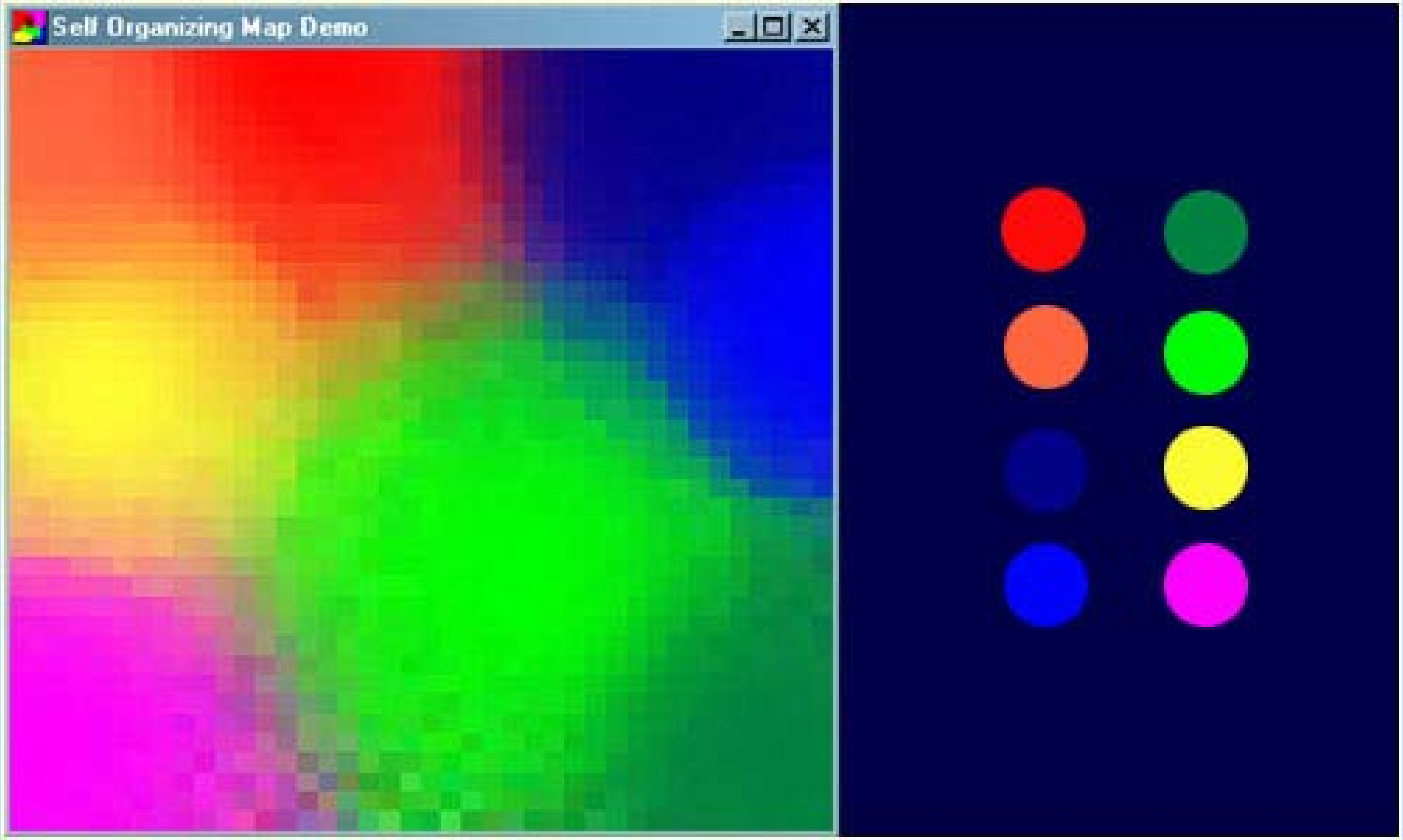
$$\Theta^* \quad = \quad \operatorname*{argmax}_{\Theta} \max_{z_1,\ldots,z_n} P_\Theta(x_1,\ldots,x_n,z_1,\ldots,z_n)$$

If we use uniform mixture probs then k-means is an application of hard EM since:

$$\log P(x,z|\theta) \propto -\sum_i d(x_i,\mu_{z(i)})^2$$

Self-Organizing Maps

Kohonen 97

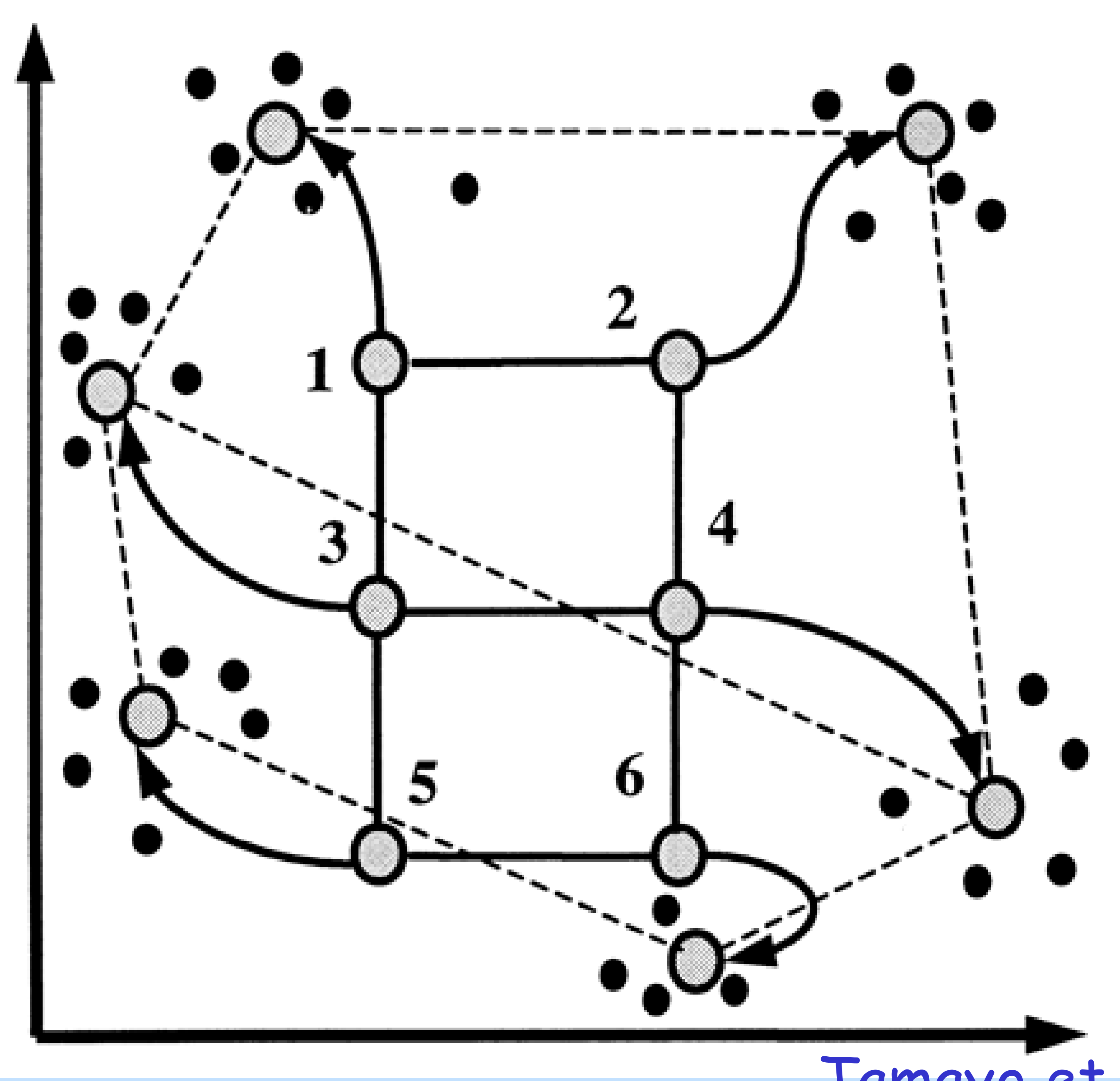


3D colors to 40x40 grid

Self-Organizing Maps

Kohonen 97

- Data: n-dim vector for each element (data point) p
- Fix a grid of k=lxm nodes; d(u,v)= dist in the grid
- Start with k arbitrary n-dim "centers" $f_0(v)$, one corresponding to each node v
- Iteration i:
  - – Pick a random data point p,
  - – Find center $f_i(v)$ closest to p
  - – Update all centers r:
    - $f_{i+1}(r) \leftarrow f_i(r) + H(v,r,i)[p-f_i(r)]$
    - H : learning function. decreases with i (iteration no.), and with d(v,r)

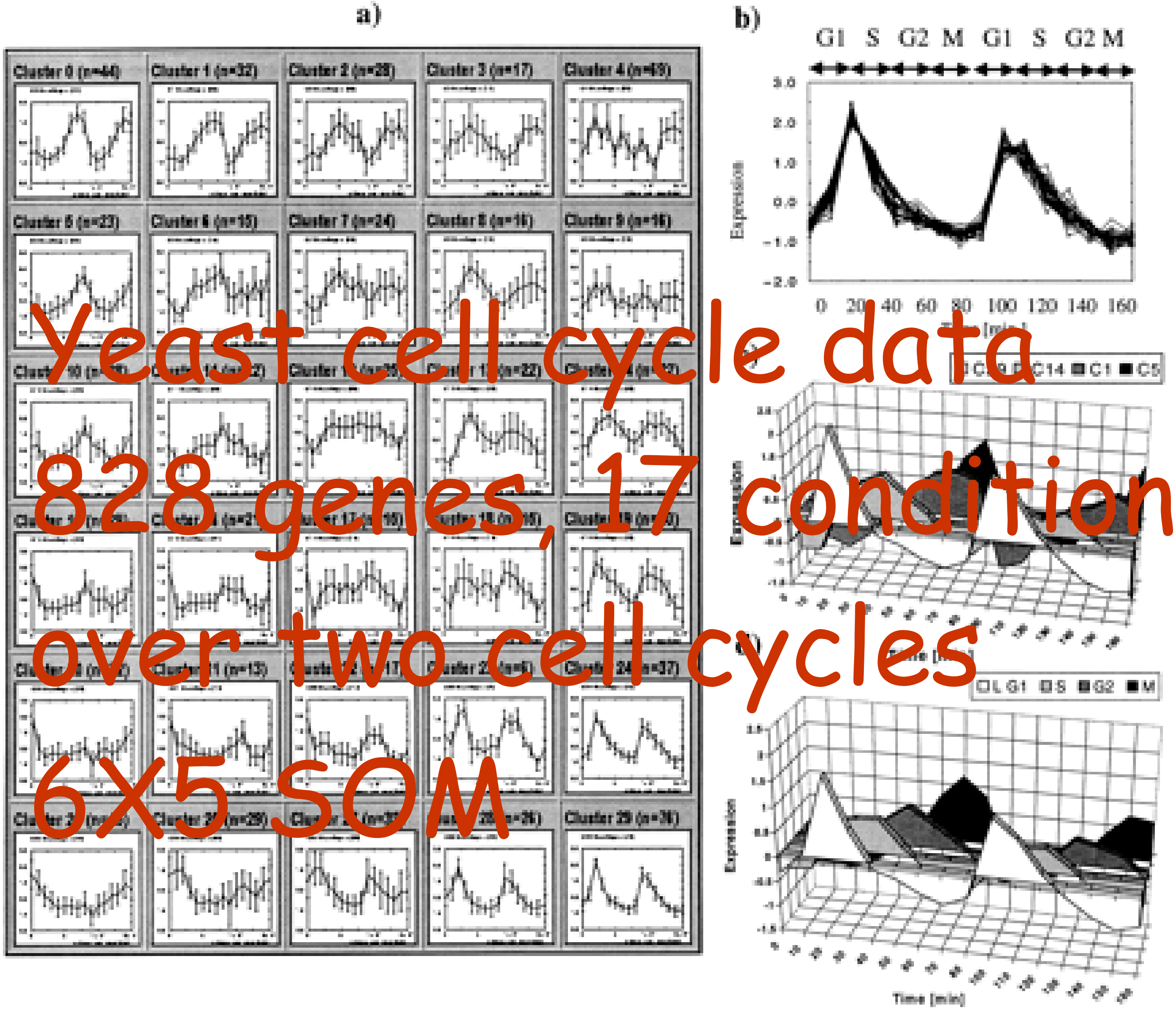


Tamayo et al, 99

GENECLUSTER

SOM software version for GE, Tamayo et al 99

- T = max no. of iterations (function of #points)
- $H(v,r,i)=0.02T/(T+100i)$ if $d(v,r)\le\rho(i)$; $=0$ o/w
- $\rho(i)$ = "radius of influence"; linearly decreasing with i, $\rho(0)=3$, $\rho(T)=0$



Yeast cell cycle data

828 genes, 17 conditions

over two cell cycles

6X5 SOM

CLICK: CLuster Identification via Connectivity Kernels

(S. & Shamir '00)

CLICK Clustering

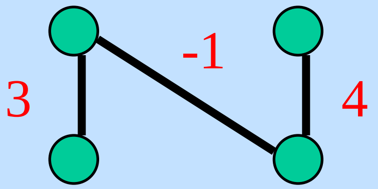
- Graph based clustering.
- Top-down: iteratively partition graph until reaching highly homogeneous subsets of elements – *kernels*.
- Greedily extend kernels by elements with similar patterns.
- The kernel identification is based on a probabilistic model of the similarity data.

Probabilistic Model

- ***Mates*** – genes that belong to the same true cluster.
- **Probabilistic assumptions**:
  - Similarity between mates $\sim N(\mu_T, \sigma_T)$
  - Similarity between non-mates $\sim N(\mu_F, \sigma_F)$
- Often observed for real data; justified in some cases by the central limit theorem.
- Parameters are estimated from partially known solutions, or using the EM algorithm.

Similarity Graph

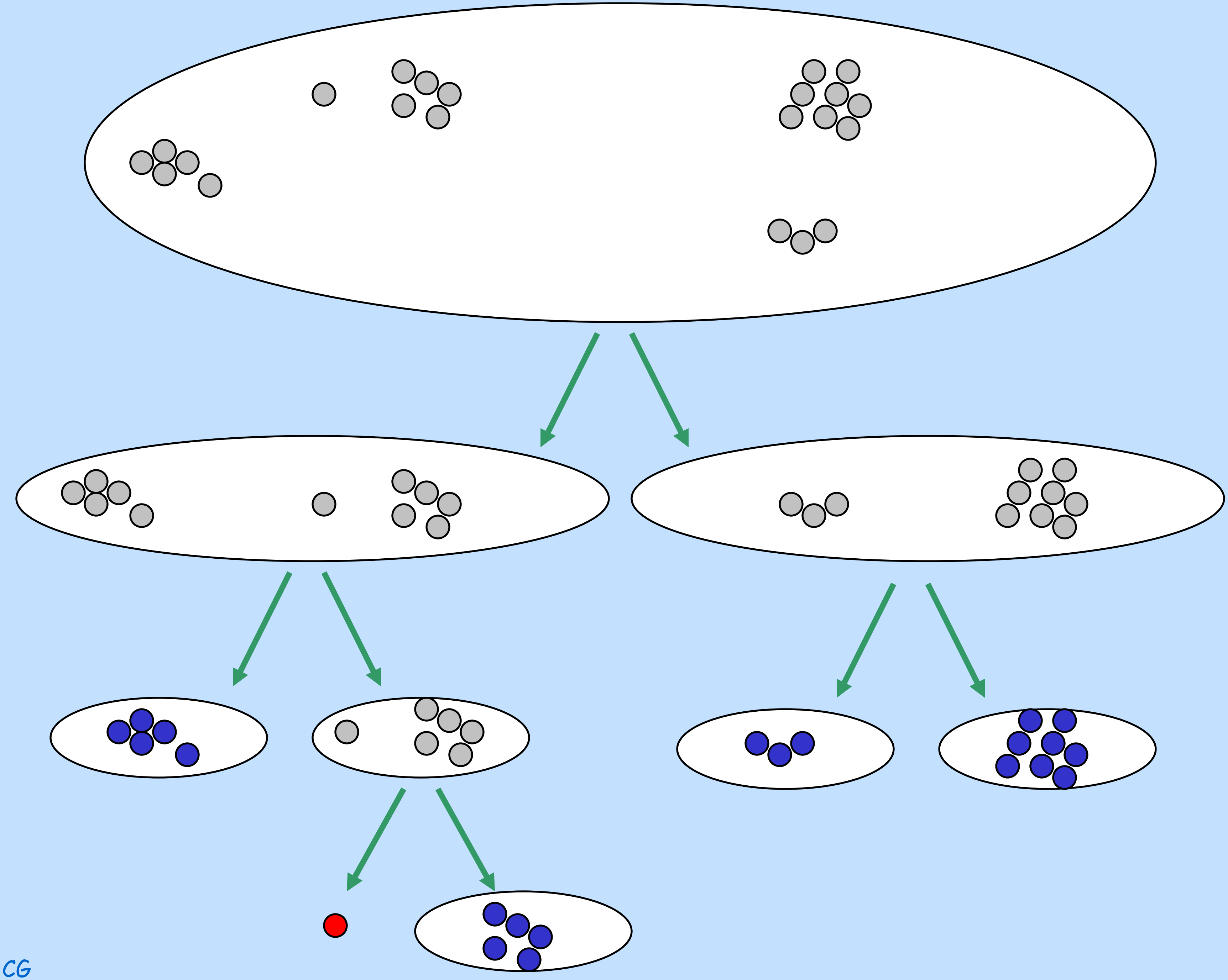
- Input ⇒ weighted graph G with a vertex per element and an edge between similar elements.



- Let $p=p_{mates}$ the fraction of mate pairs. Define edge weights to reflect prob. of mates:

$$w_{ij} = \ln \frac{\Pr(i, j \text{ are mates} \mid S_{ij})}{\Pr(i, j \text{ are non-mates} \mid S_{ij})} =$$

$$\ln \frac{p\sigma_F}{(1-p)\sigma_T} + \frac{(S_{ij}-\mu_F)^2}{2\sigma_F^2} - \frac{(S_{ij}-\mu_T)^2}{2\sigma_T^2}$$

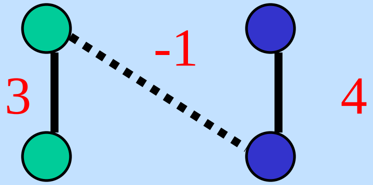


CG

Kernel Identification

Cut - Partition of vertices into two groups.

Weight - Sum of weights across the cut.



- For each cut C in G we test two hypotheses:

H_0: C contains only edges between non-mates.

H_1: C contains only edges between mates.

G is declared a **kernel** if H_1 is more probable for all cuts.

Kernel Identification

<u>Thm</u>: G is a kernel iff weight of min. cut > 0.

Proof: By Bayes thm., for any cut C:

$$\log \frac{\Pr(H_1^C|C)}{\Pr(H_0^C|C)} = \log \frac{\Pr(H_1^C) f(C|H_1^C)}{\Pr(H_0^C) f(C|H_0^C)}$$

$$= |C| \log \frac{p_{\text{mates}}\sigma_F}{(1-p_{\text{mates}})\sigma_T} + \sum_{(i,j)\in C} \frac{(S_{ij}-\mu_F)^2}{2\sigma_F^2}$$

$$- \sum_{(i,j)\in C} \frac{(S_{ij}-\mu_T)^2}{2\sigma_T^2} = W(C).$$

In particular, if H_1 is more probable for the min. cut C, then this is true for any other cut C’, since:

$$\log \frac{\Pr(H_1^C|C)}{\Pr(H_0^C|C)} = W(C) \le W(C') = \log \frac{\Pr(H_1^{C'}|C')}{\Pr(H_0^{C'}|C')}.$$

Kernel Identification Algorithm

```
Basic-CLICK(G=(V,E)):
If V={v} then mark v as a **singleton**.
Else if G is a **kernel** then
    Output V.
Else
    (A,B)←Min-Weight-Cut(G).
    Basic-CLICK(A).
    Basic-CLICK(B).
```

Refinements

- **Adoption Step:** Find kernel *K* and singleton *s* with highest similarity. Adopt *s* to *K* if that similarity is sufficiently high.

⇨ Iterative application of Kernel Identification and the adoption step.

- **Merging Step: (**at the end of the algorithm) Greedily merge clusters whose average patterns are sufficiently similar.

- **Min-cut:** NPC when negative weights; heuristic: compute ignoring neg. weight edges and then correct weight for the kernel test.

CLICK Simulation: Setup

- Cluster structures: 6*50, 10*30 and 10,…,80.
- Mates similarity ~ $N(\mu_T,\sigma)$
- Non-mates similarity ~ $N(\mu_F,\sigma)$
- $\sigma=5$
- $\mu_T - \mu_F = t\sigma$, t= 2, 1, .8, .6

CLICK Simulation Results

Mean Jaccard score over 20 runs.

Distance (stds) Structure	2	1	0.8	0.6
6 * 50	1	1	0.98	0.85
10 * 30	1	0.96	0.71	0.1
10,…,80	1	1	0.97	0.83

Quality Assessment

no correct clustering is known

Homogeneity: average similarity between mates; minimum cluster homogeneity.

$$H_{avg} = \frac{2}{\sum_{C} |C|\left(|C|-1\right)} \sum_{C} \sum_{i,j \in C} S(i,j)$$

Separation: average similarity between non-mates; maximum inter-cluster similarity.

$$S_{avg} = \frac{1}{\sum_{C<C'} |C||C'|} \sum_{C<C'} \sum_{i \in C, j \in C'} S(i,j)$$

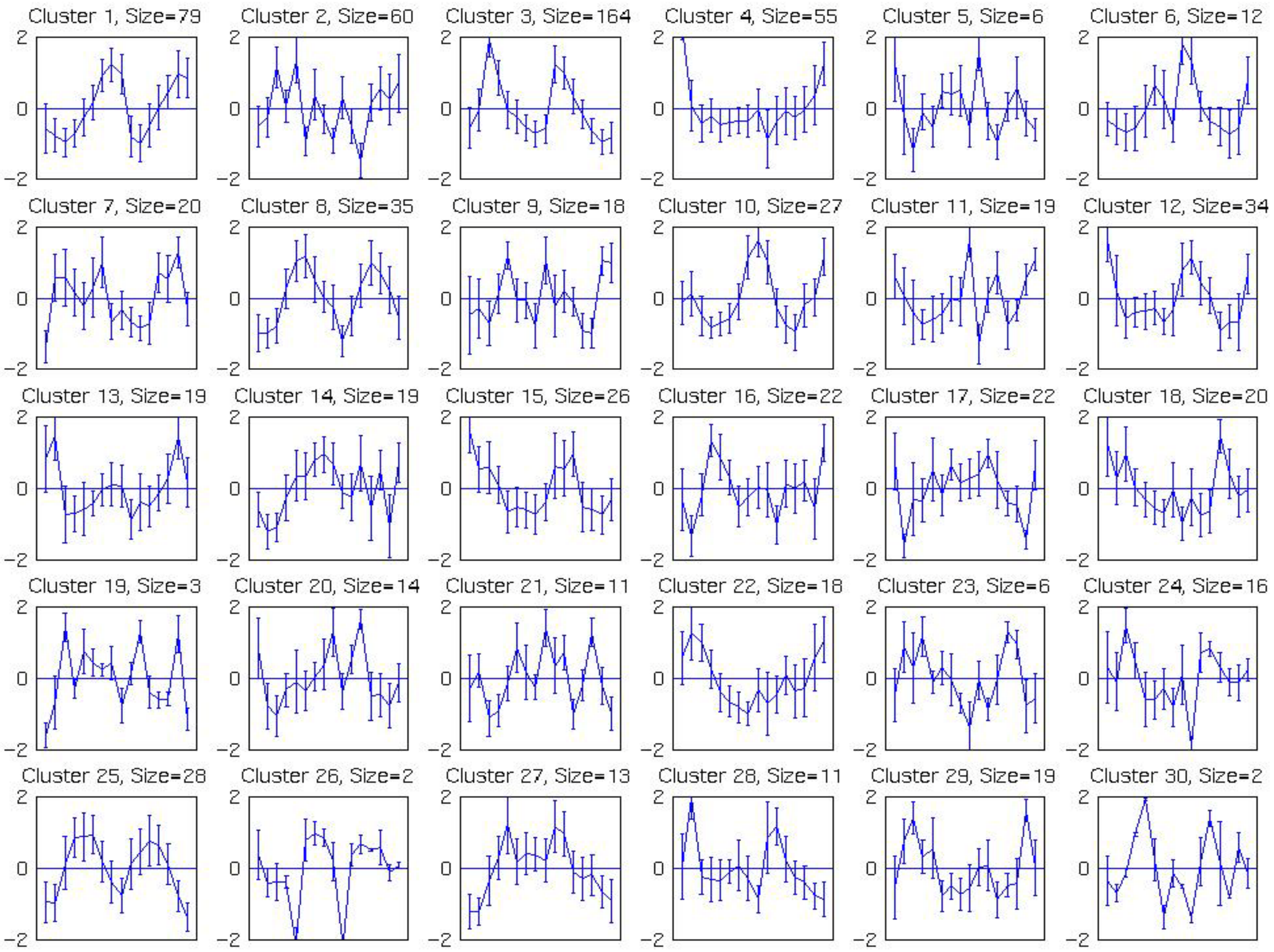
Gene Expression: Yeast Cell Cycle

Expression levels of 826 yeast genes, measured at 16 time points over two cell cycles (Cho et al. 1998).

	Clusters	Homogeneity ⇧		Separation ⇩	
		Ave	Min	Ave	Max
CLICK	30	0.8	-0.19	-0.07	0.65
GeneCluster*	30	0.74	-0.88	-0.02	0.97

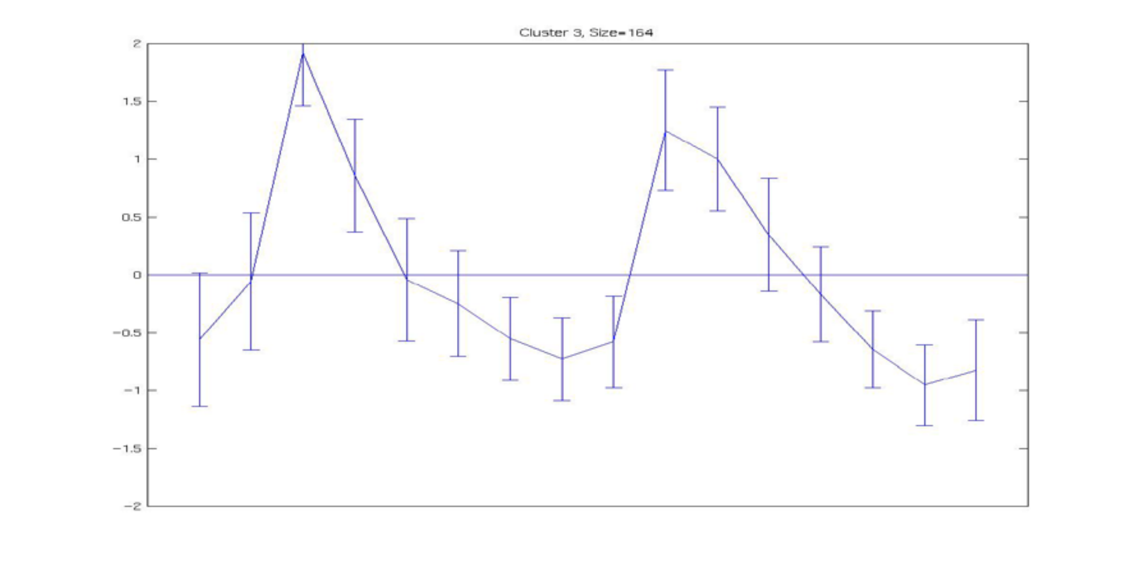
*Tamayo et al. 1999.

CLICK clusters: Yeast Cell Cycle



Yeast Cell Cycle: late G1 Cluster

N=164



- Contains 91% of late G1-peaking genes.
- In contrast, in GeneCluster 87% are split among 3 clusters.

Gene Expression: Serum Response

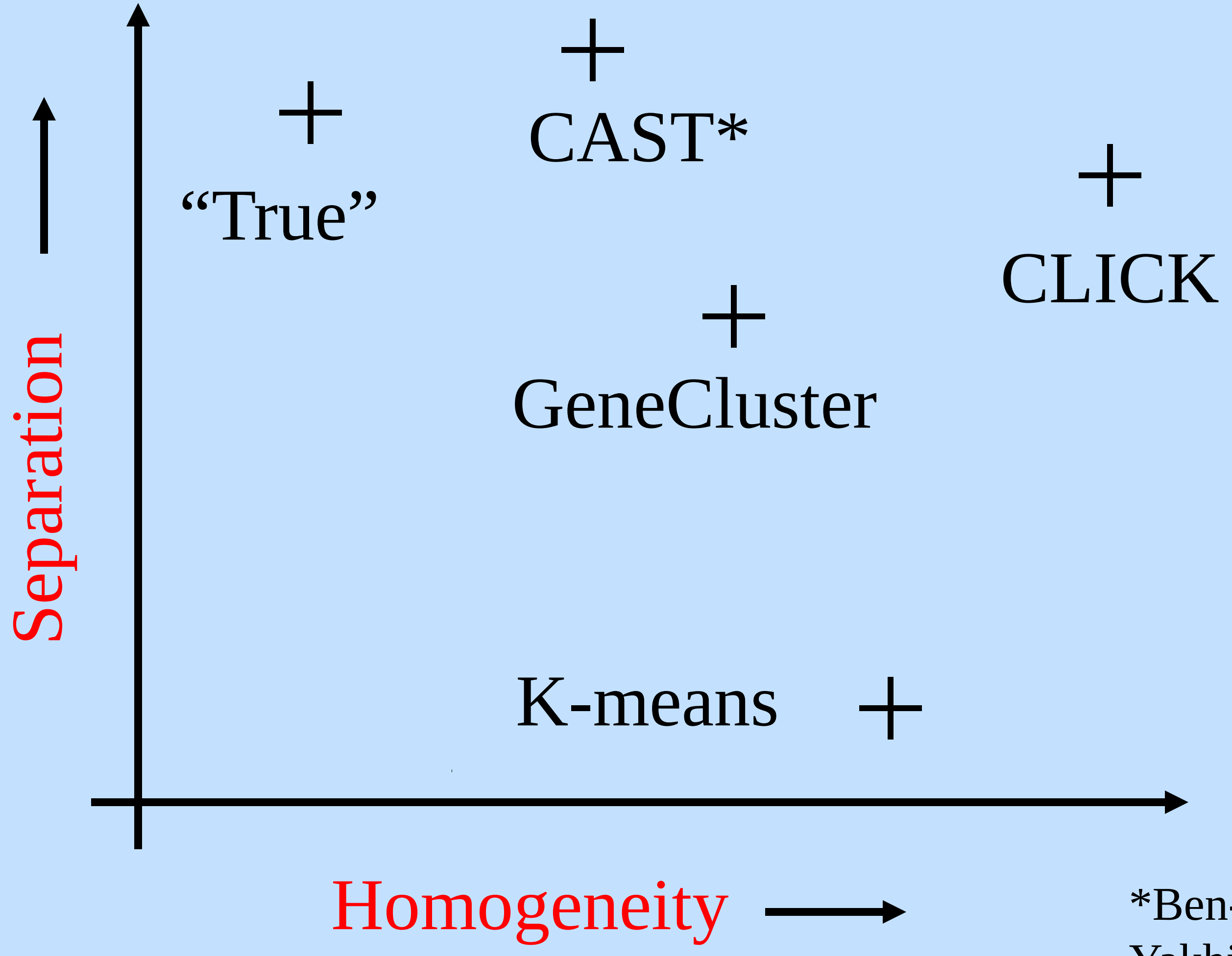
Human fibroblast cells starved for 48 hours, then stimulated by serum. Expression levels of 8,613 genes measured at 13 time points. (Data from Iyer et al., *Science* 1999)

	Clusters	Homogeneity ↑		Separation ↓	
		Ave	Min	Ave	Max
CLICK	**10**	**0.88**	**0.13**	**-0.34**	**0.65**
CLUSTER*	**10**	**0.87**	**-0.75**	**-0.13**	**0.9**

* Eisen et al., *PNAS* 1998.

Performance on Yeast Cell Cycle Data

698 genes, 72 conditions (Spellman et al. 1998). Each algorithm was run by its authors in a “blind” test.



*Ben-Dor, Shamir, Yakhini ‘99

EXPression ANalyzer and DisplayER

analysis and visualization tool for gene expression data, including:

Clustering CLICK, KMeans, SOM, hierarchical	**Promoter analysis** PRIMA	**Biclustering** SAMBA
Functional enrichment	**Visualization**	

Software is available at
http://www.cs.tau.ac.il/~rshamir