

Cluster Analysis for Gene Expression Data

CSE 5615 Class Project, Group 2

Stephen Connetti

Sally Ellingson

Luis Quiles

Gene Expression Analysis?

- Virus, Bacteria, and Cellular Function Research
- Map unknown genes to cellular functions
- Map illness to malfunctioning cellular processes
- Measuring gene expression – comparing expression of proteins under different situations against a base situation

Gene Expression - Data

- To measure gene expression, measure the mRNA being produced vs. base production
- Data: mRNA production vs. microarray test

	E. Coli	E. coli	E. coli	E. coli	EHEC	EHEC	EHEC	EHEC	EHEC
Gene	1 hr	6 hr	12 hr	24 hr	1 hr	2 hr	6 hr	12 hr	24 hr
GCSF	0.083	2.615	2.007	1.96	0.001	0.714	3.642	3.138	2.229
GMCSF	0.722	2.002	0.940	1.21	1.034	1.430	2.961	2.920	2.352
IL12B	0.845	4.77	4.369	3.454	0.426	-0.426	4.316	4.816	3.671
IL1RN	0.548	1.732	1.938	1.389	0.781	-1.27	1.344	1.419	1.301
IL6	3.006	5.244	3.897	3.957	3.889	4.106	4.396	4.137	3.990

Gene Expression - Clustering

- Groups together data with similar properties
- Data sets are partitioned – clusters contain points more similar to themselves than others
- Clustering aids researchers to infer relationships between genes, especially when cellular functions are known
- Also helps identify relationships in co-expressed genes

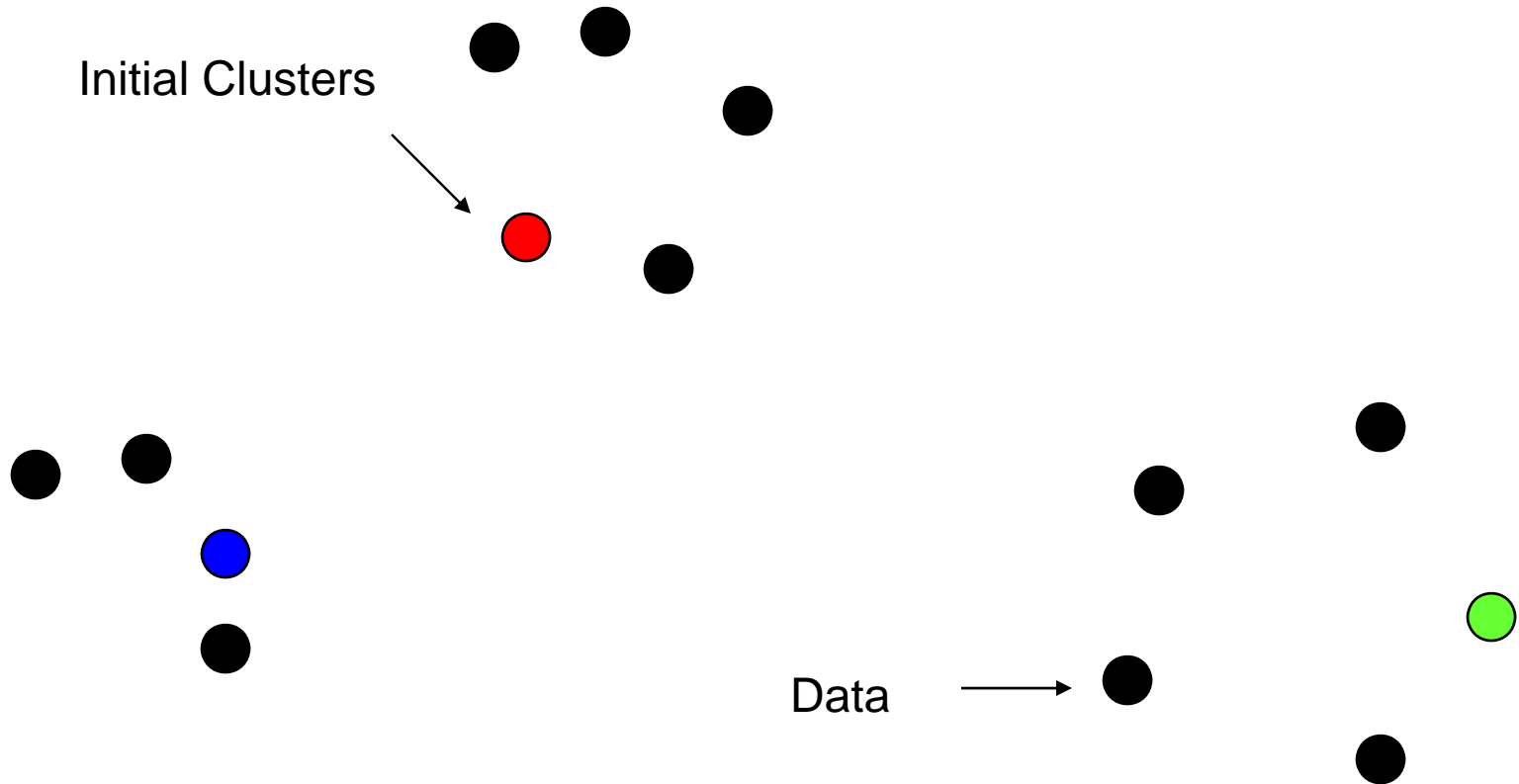
Human Macrophage Activation Programs induced by Pathogens

- Originating source of our data
- 6800 genes, 43 microarray tests
- Significance tests reduce this to 977 genes
- Results were clustered to find relevance
- 198 genes expressed with the same pattern

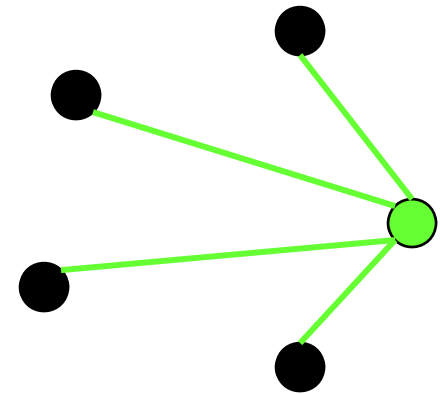
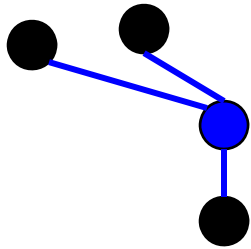
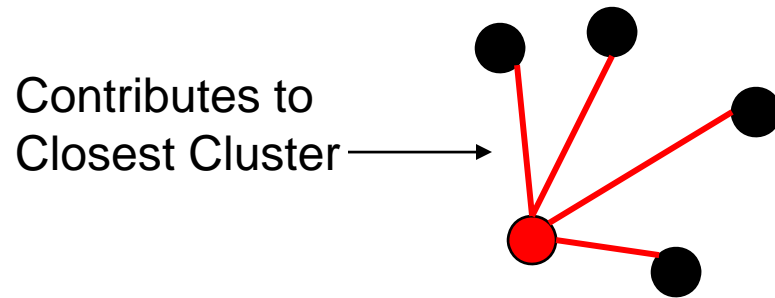
Project Goals

- Implement 3 Clustering algorithms
 - Bayesian Clustering, Self-Organizing Maps, CLICK
- Find most similar clusters – relevant similarity
- Compare paper's best cluster to our results
 - How many of the 198 genes did we find?

K-Means

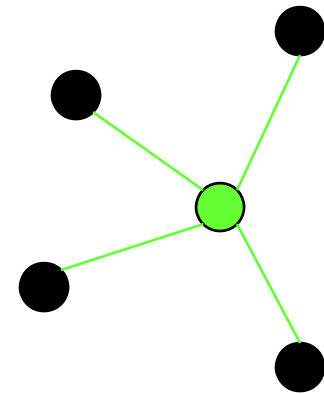
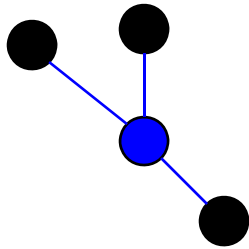
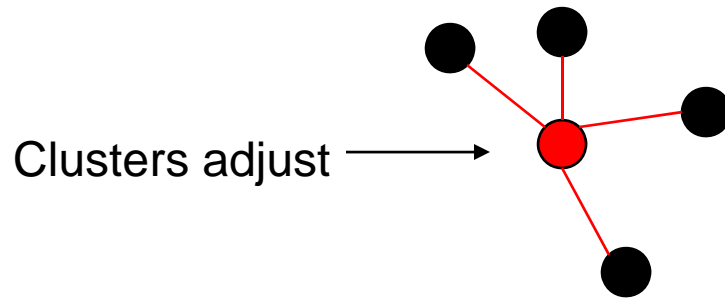


K-Means

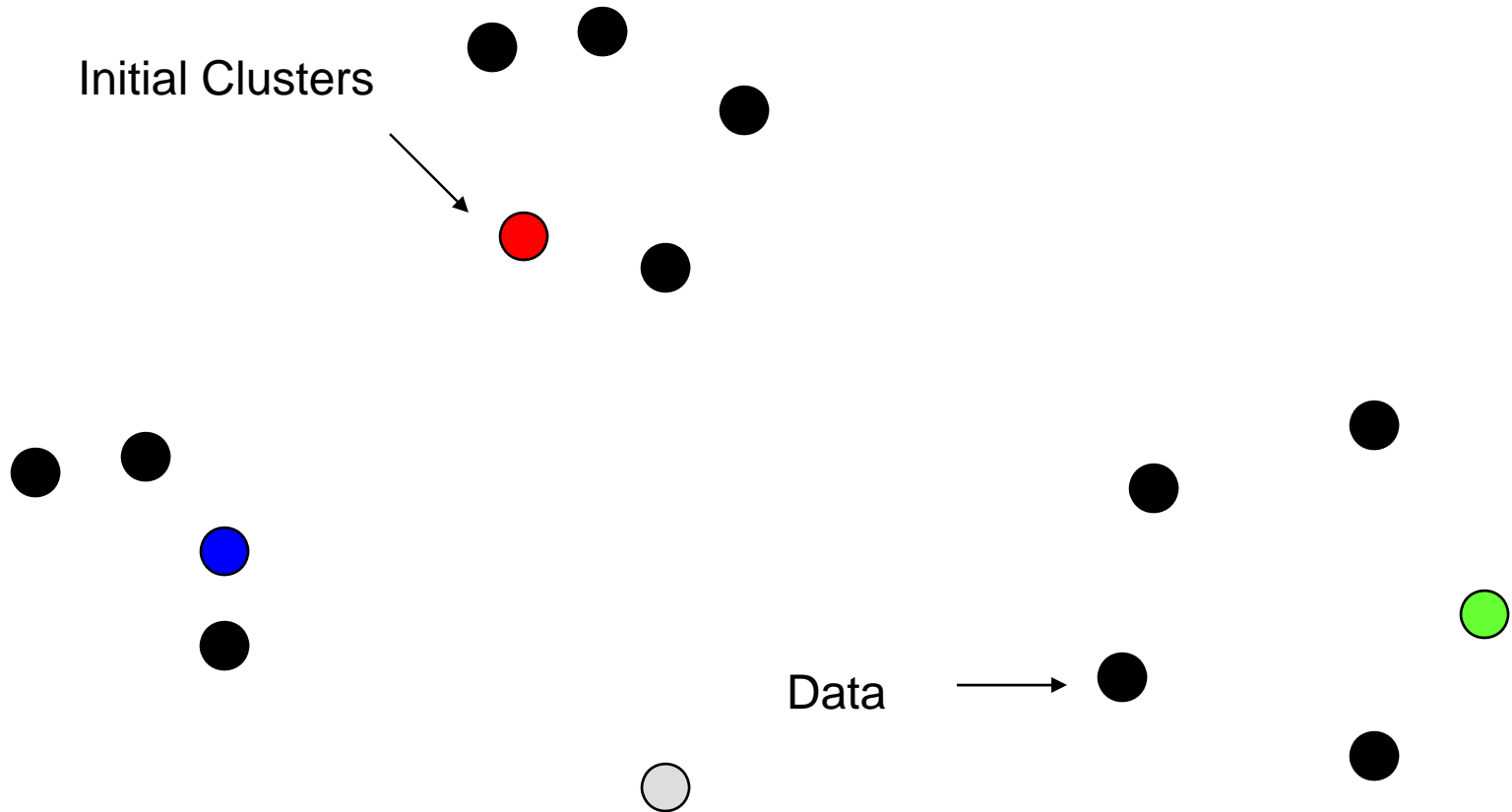


[3.2] Fraley, C., and Raftery, A. How Many Cluster? Which Clustering Method? Answers Via Model-Based Cluster Analysis.

K-Means

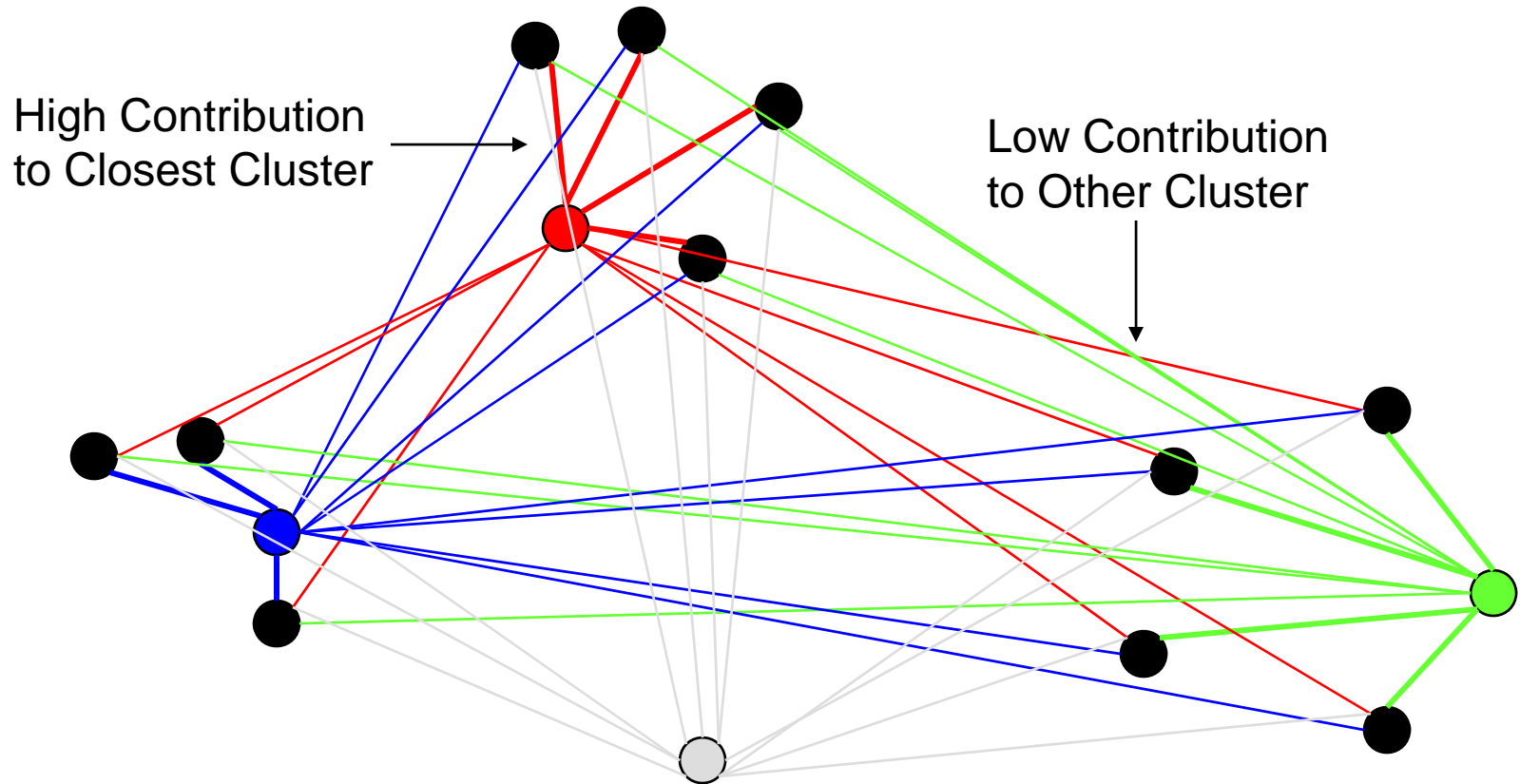


Bayesian Clustering



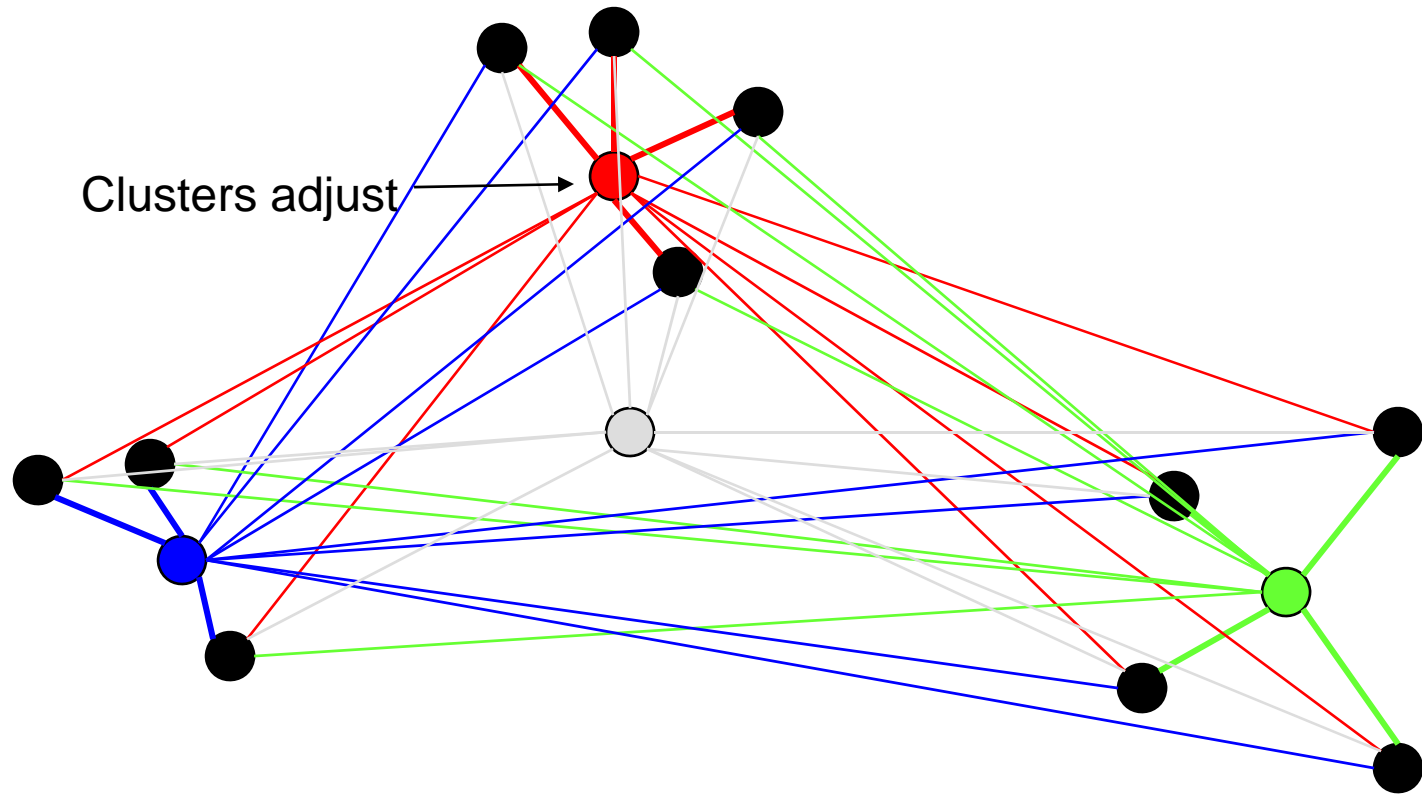
[3.2] Fraley, C., and Raftery, A. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis.

Bayesian Clustering



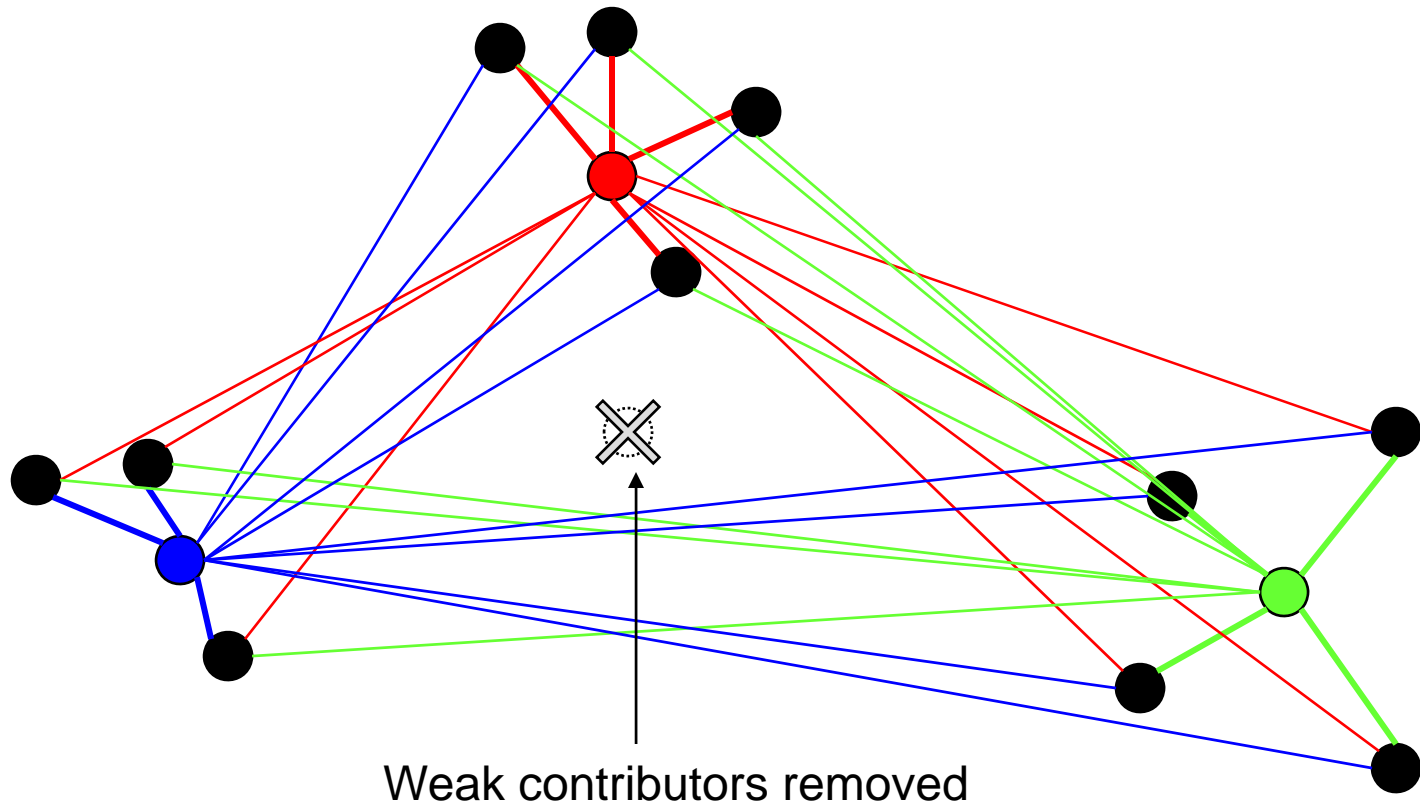
[3.2] Fraley, C., and Raftery, A. How Many Cluster? Which Clustering Method? Answers Via Model-Based Cluster Analysis.

Bayesian Clustering



[3.2] Fraley, C., and Raftery, A. How Many Cluster? Which Clustering Method? Answers Via Model-Based Cluster Analysis.

Bayesian Clustering



[3.2] Fraley, C., and Raftery, A. How Many Cluster? Which Clustering Method? Answers Via Model-Based Cluster Analysis.

Bayesian Clustering

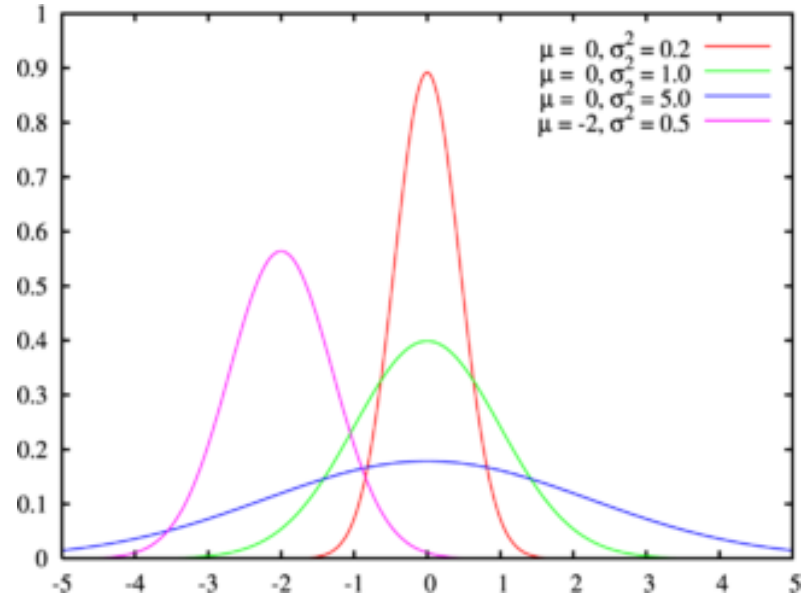
- Initialize Clusters
- Until Convergence
 - Expectation: Compute new probabilities
 - Maximization: Compute new clusters
- Use BIC to prune clusters
- Compute global BIC
- Repeat with different initial clusters
 - Keep results with best BIC

Normal Distribution

$$N(x) = \frac{e^{-(x-x_0)^2 / 2\sigma^2}}{\sigma\sqrt{2\pi}}$$

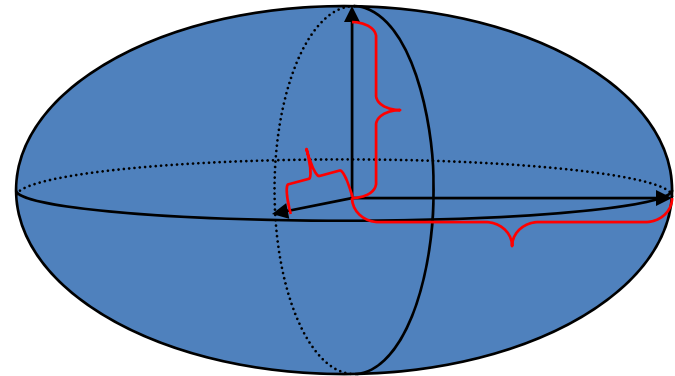
$$N(\vec{x}) = \frac{e^{-(\vec{x}-\vec{\mu})\Sigma^{-1}(\vec{x}-\vec{\mu})/2}}{|\Sigma| \sqrt{(2\pi)^n}}$$

$$P(\vec{d}_i \in \mu_k^t | \mu_k^t) = \frac{e^{-(\vec{d}_i - \vec{\mu}_k^t)\Sigma_k^{-1}(\vec{d}_i - \vec{\mu}_k^t)/2}}{|\Sigma_k| \sqrt{(2\pi)^n}}$$



Expectation : Non-spherical Clusters

$$\Sigma_k = D_k \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & 0 \\ \cdot & & \ddots & \cdot \\ \cdot & & & \cdot \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} D_k^T$$



- D_k =Eigenvectors of Σ are the orientation of cluster k.
- $\lambda_1 \dots \lambda_n$ =Eigenvalues of Σ are the radii of cluster k.
- Spherical clusters are usually sufficient (Identity Matrix)

Compute initial probabilities


- Given the initial clusters initialize the probabilities with the Normal Distribution.

$$P(\vec{d}_i \in \mu_k^0 \mid \mu_k^0) = e^{-(\vec{d}_i - \mu_k^0) \cdot (\vec{d}_i - \mu_k^0) / 2}$$

- Since the probabilities will have to be normalized anyway, we can skip the constant.

Expectation: Baye's law

Posterior Prior


$$P(Cause | Effect) = \frac{P(Effect | Cause) \cdot P(Cause)}{P(Effect)}$$

$$P(Cause | Effect) = \frac{P(Effect | Cause) \cdot P(Cause)}{\sum P(Effect | Cause) \cdot P(Cause)}$$

Expectation: Baye's law

New Probability

Old Probabilities

$$P(\vec{\mu}_k^t | \vec{d}_i \in \vec{\mu}_k^t) = \frac{P(\vec{d}_i \in \vec{\mu}_k^t | \vec{\mu}_k^t) \cdot \sum_{i=1}^n P(\vec{\mu}_k^{t-1} | \vec{d}_i \in \vec{\mu}_k^{t-1})}{\sum_{j=1}^C P(\vec{d}_i \in \vec{\mu}_j^t | \vec{\mu}_j^t) \cdot \sum_{i=1}^n P(\vec{\mu}_j^{t-1} | \vec{d}_i \in \vec{\mu}_j^{t-1})}$$

Recompute membership probabilities using normal distribution.

Maximization: New Clusters

- Clusters are computed as the weighted average of each point and the probability it belongs to the cluster.

$$\vec{\mu}_k^t = \frac{\sum_{i=1}^n P(\vec{\mu}_k^{t-1} \mid \vec{d}_i \in \vec{\mu}_k^{t-1}) \cdot \vec{d}_i}{\sum_{i=1}^n P(\vec{\mu}_k^{t-1} \mid \vec{d}_i \in \vec{\mu}_k^{t-1})}$$

Bayesian Information Criterion

- BIC measures the efficiency of the parameterized model in terms of predicting the data.
- It is independent to prior knowledge and the model used.

$$BIC_k^t = n \cdot \ln \left(\frac{\sum_{i=1}^n P(\vec{d}_i \in \vec{\mu}_k^t | \vec{\mu}_k^t)^2}{n} \right) + k \cdot \ln(n)$$

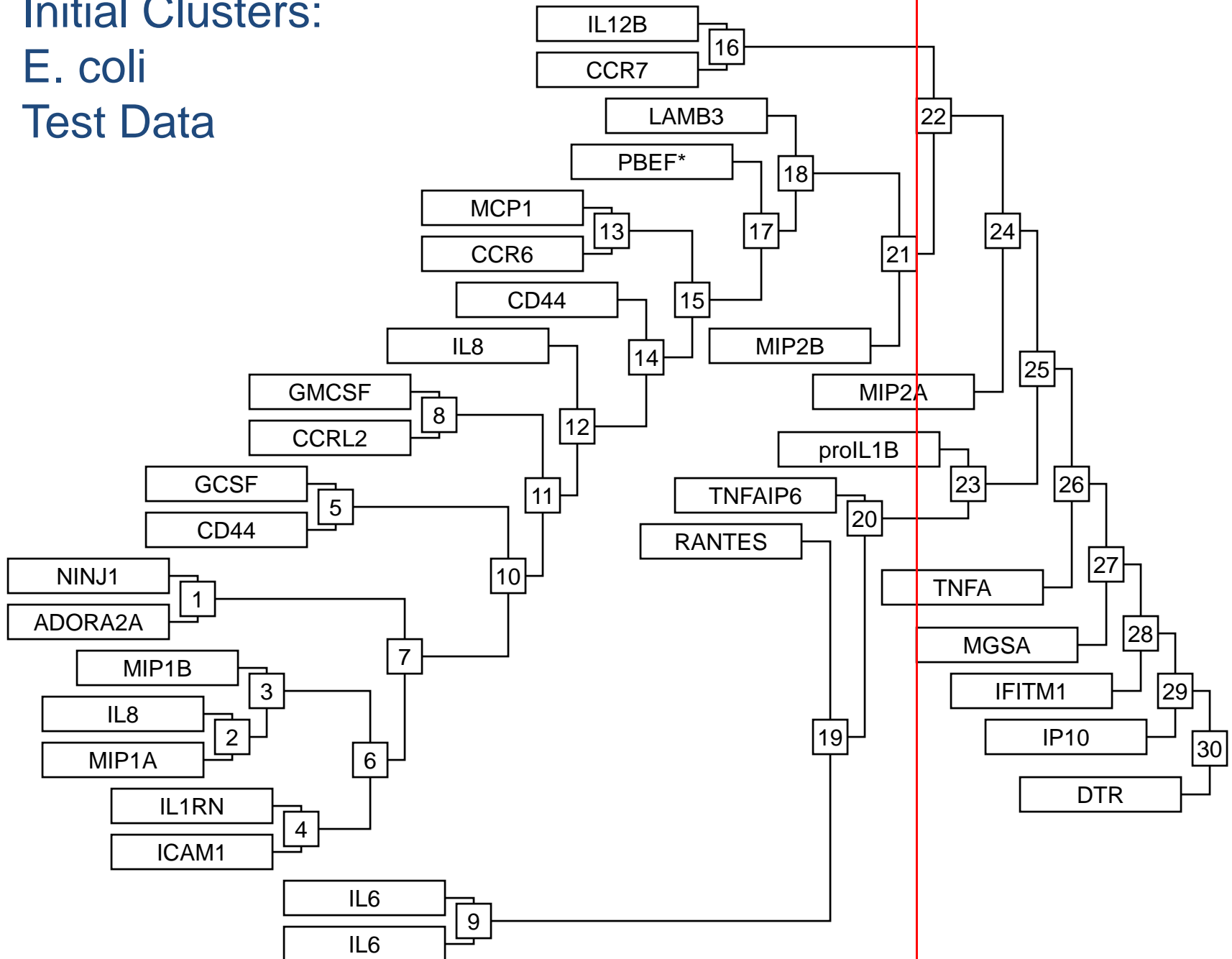
Fit quality Complexity penalty

Bayesian Information Criterion

- After convergence compute global BIC.

$$BIC = n \cdot \ln \left(\frac{\sum_{i=1}^n \sum_{j=1}^k P(\vec{d}_i \in \vec{\mu}_j | \vec{\mu}_j)^2}{nk} \right) + k \cdot \ln(n)$$

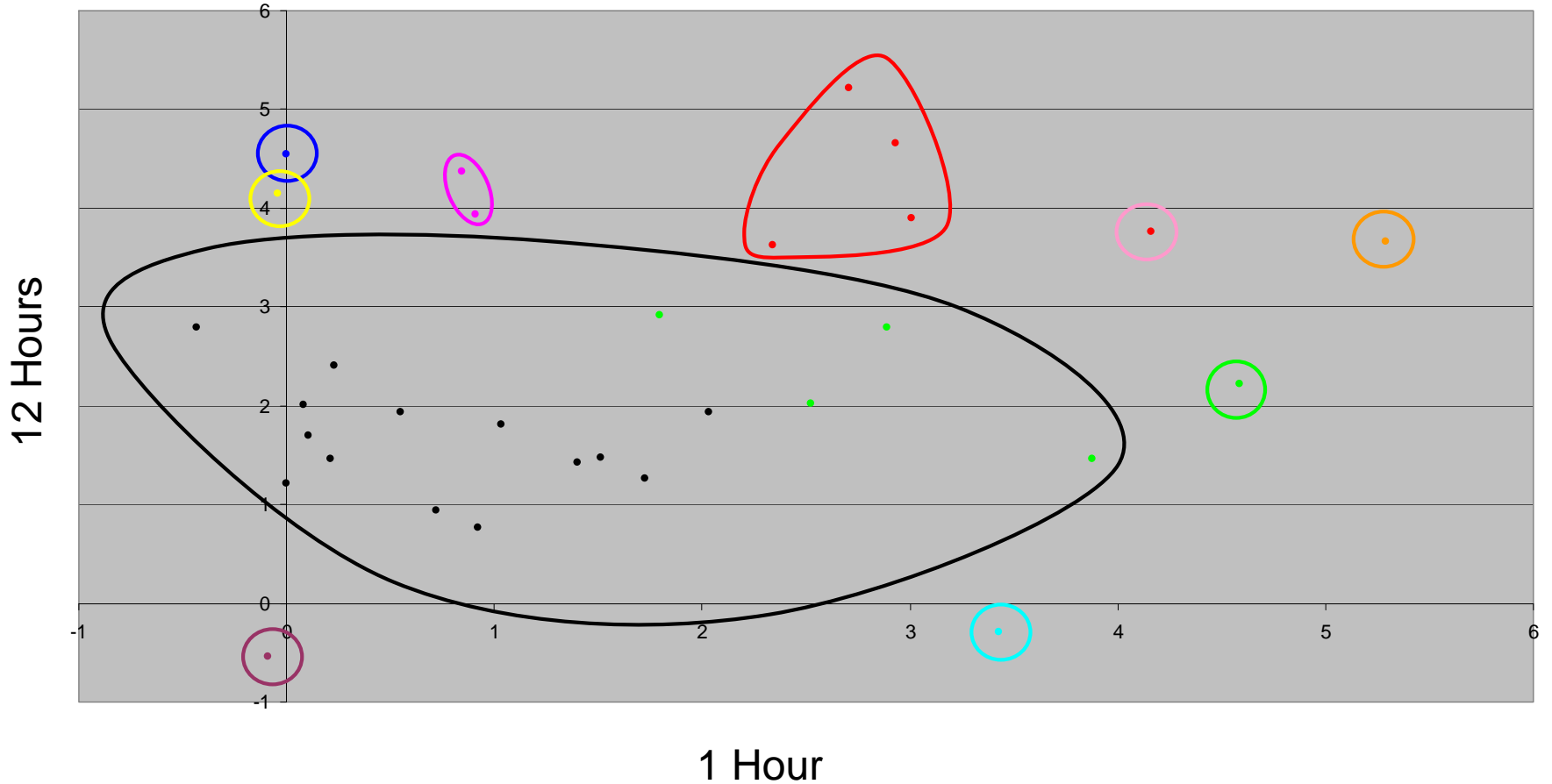
Initial Clusters: E. coli Test Data



E-Coli Initial Clusters (Test Data)

Hour 1 vs Hour 12

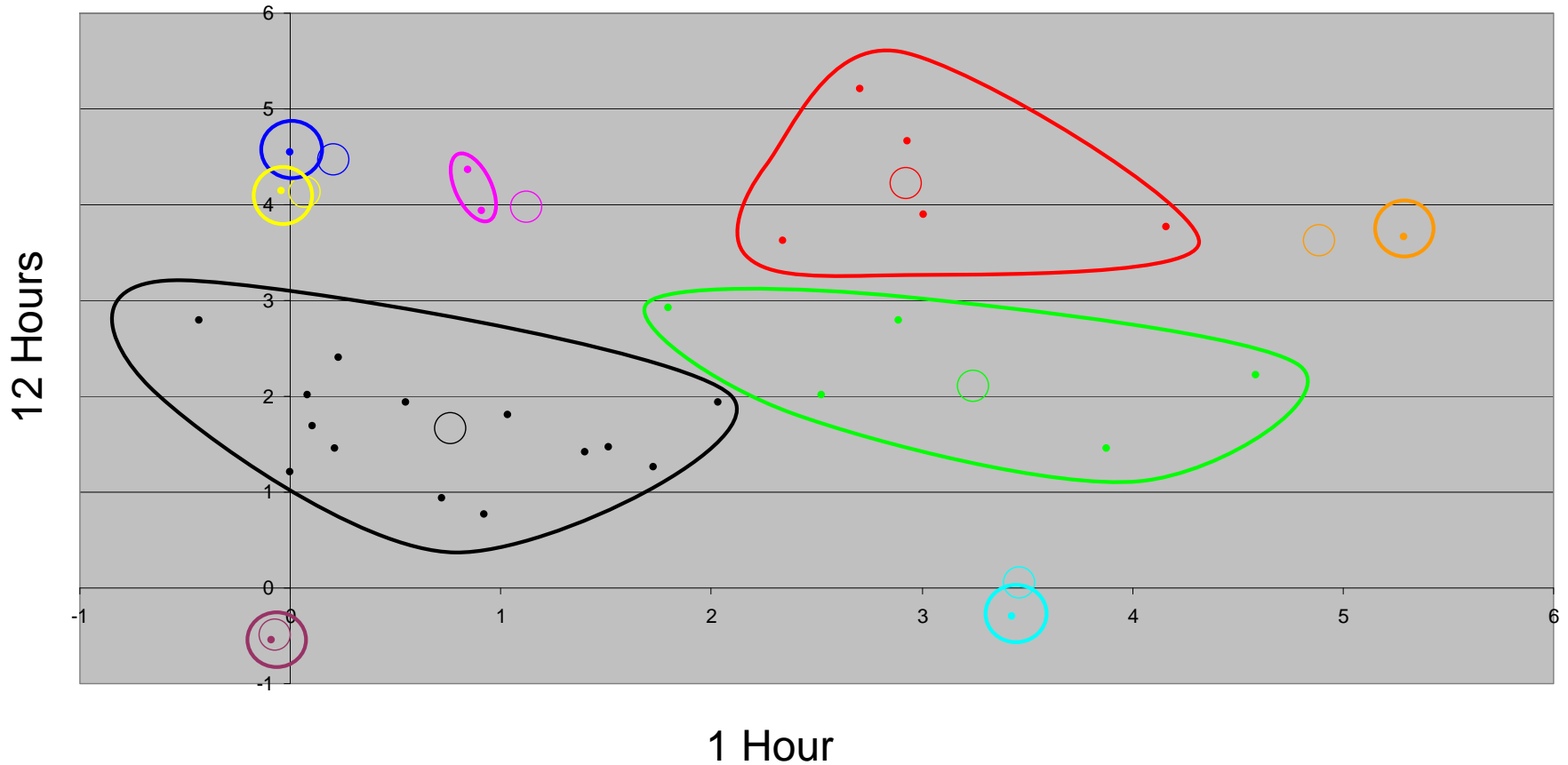
• GCSF	• GMCSF	• IL12B	• IL1RN	• IL6	• IL6	• PBEF*	• proIL1B	• TNFA	• IL8	• IL8
• IP10	• MCP1	• MGSA	• MIP1A	• MIP1B	• MIP2A	• MIP2B	• RANTES	• CD44	• CD44	• ICAM1
• IFITM1	• LAMB3	• NINJ1	• TNFAIP6	• ADORA2A	• CCR6	• CCR7	• CCRL2	• DTR		



E-Coli Final Clusters (Test Data)

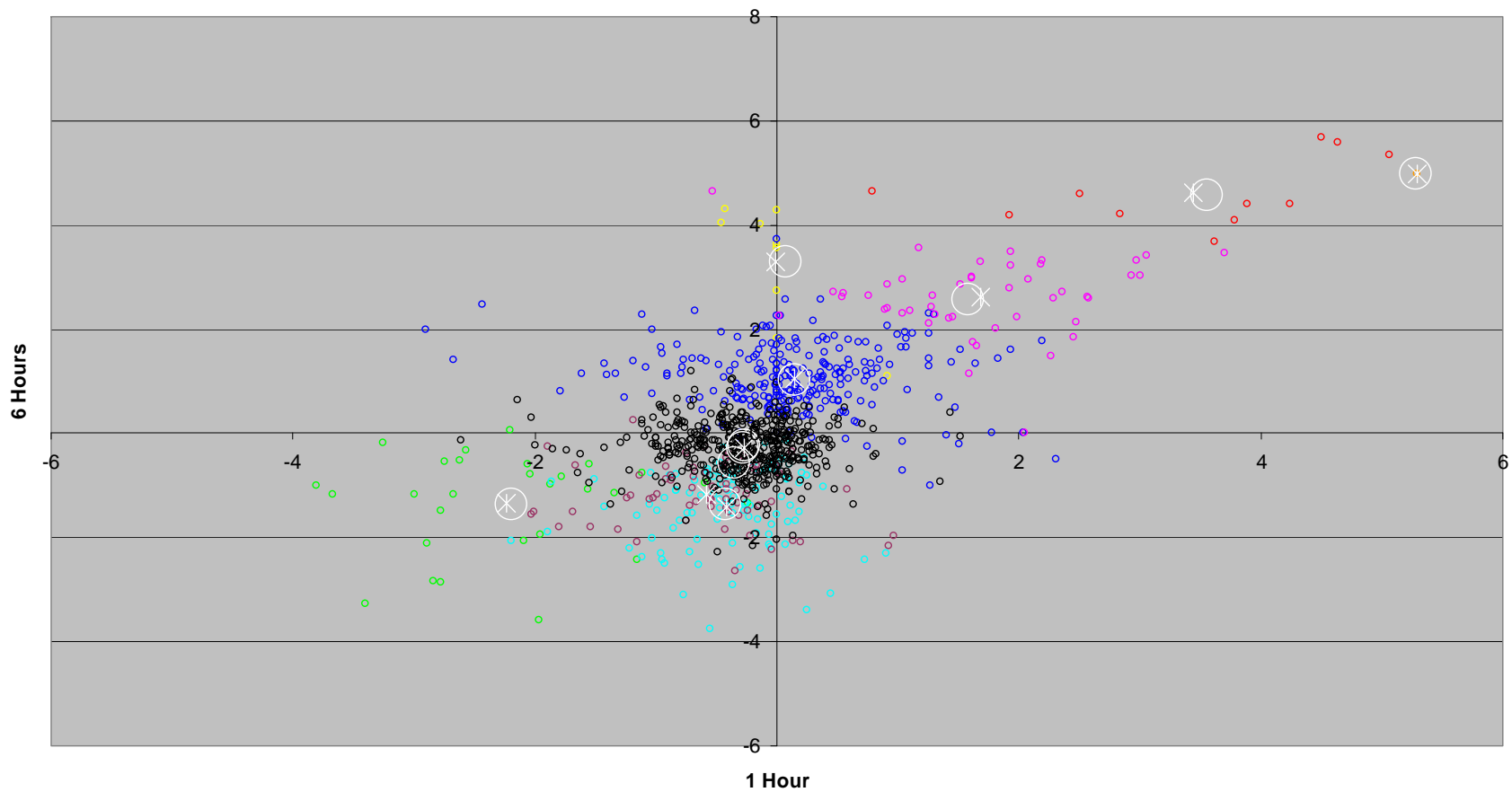
Hour 1 vs Hour 12

- | | | | | | | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------|--------|-------------|-------------|
| • GCSF | • GMCSF | • IL12B | • IL1RN | • IL6 | • IL6 | • PBEF* | • proIL1B | • TNFA | • IL8 | • IL8 |
| • IP10 | • MCP1 | • MGSA | • MIP1A | • MIP1B | • MIP2A | • MIP2B | • RANTES | • CD44 | • CD44 | • ICAM1 |
| • IFITM1 | • LAMB3 | • NINJ1 | • TNFAIP6 | • ADORA2A | • CCR6 | • CCR7 | • CCRL2 | • DTR | ○ Cluster 1 | ○ Cluster 2 |
| ○ Cluster 3 | ○ Cluster 4 | ○ Cluster 5 | ○ Cluster 6 | ○ Cluster 7 | ○ Cluster 8 | ○ Cluster 9 | | | | |



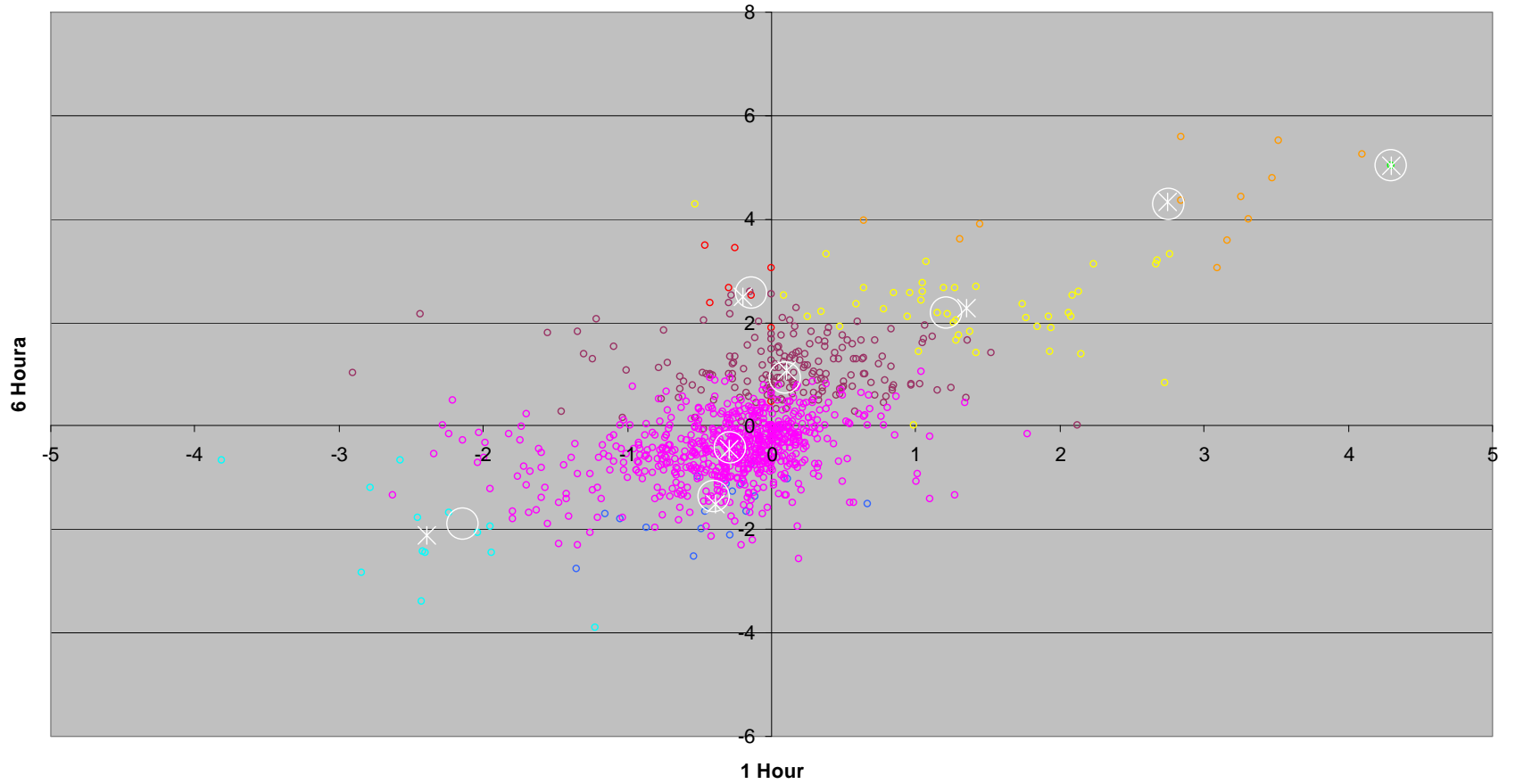
EHEC

Cluster 0 Cluster 1 Cluster 2 Cluster 3 Cluster 4 Cluster 5 Cluster 6 Cluster 7 Cluster 9 Rational Clusters Discrete Clusters



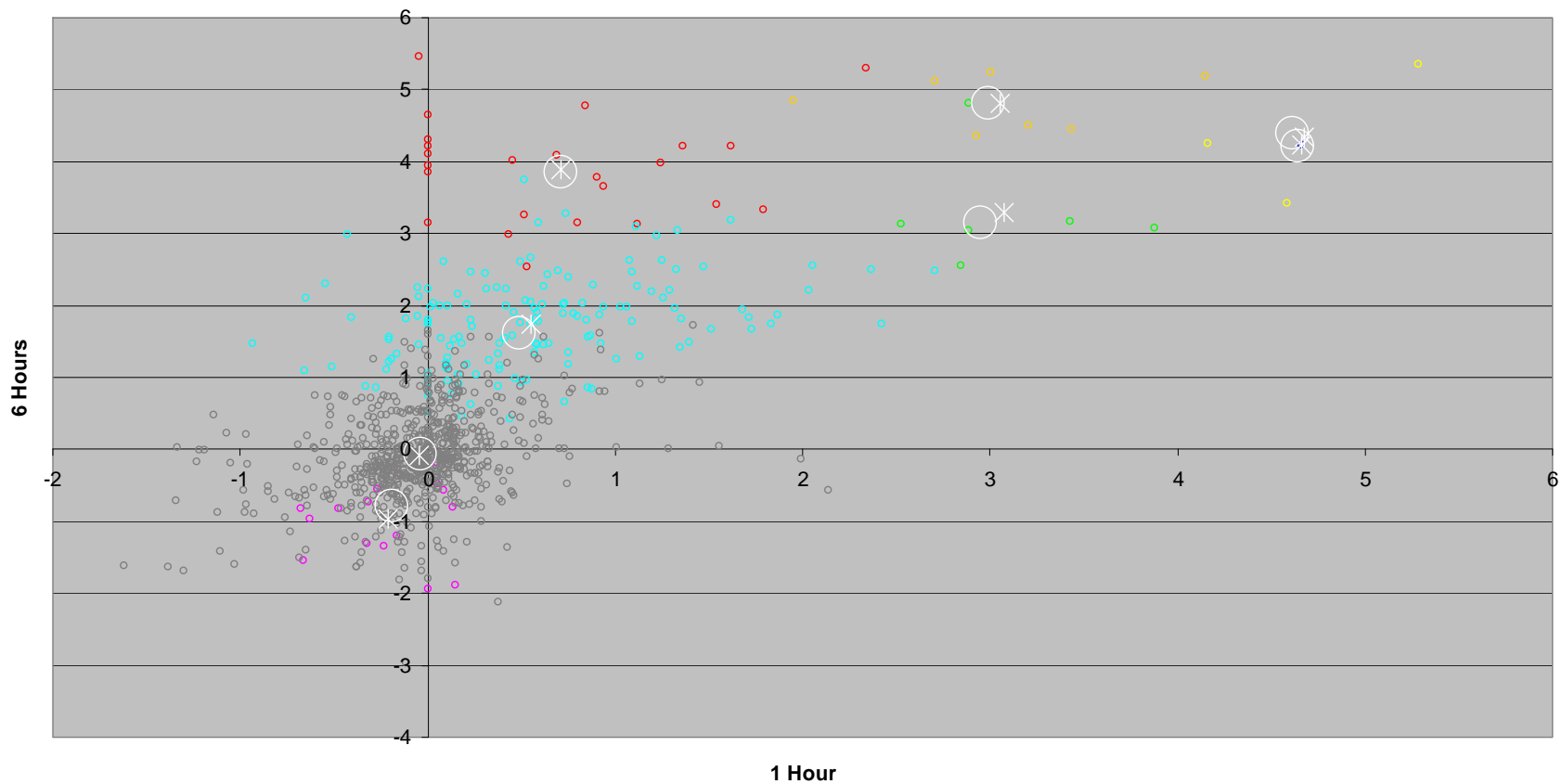
S Aureus

Cluster 0 Cluster 1 Cluster 2 Cluster 3 Cluster 4 Cluster 5 Cluster 6 Cluster 7 Rational Clusters Discrete Clusters



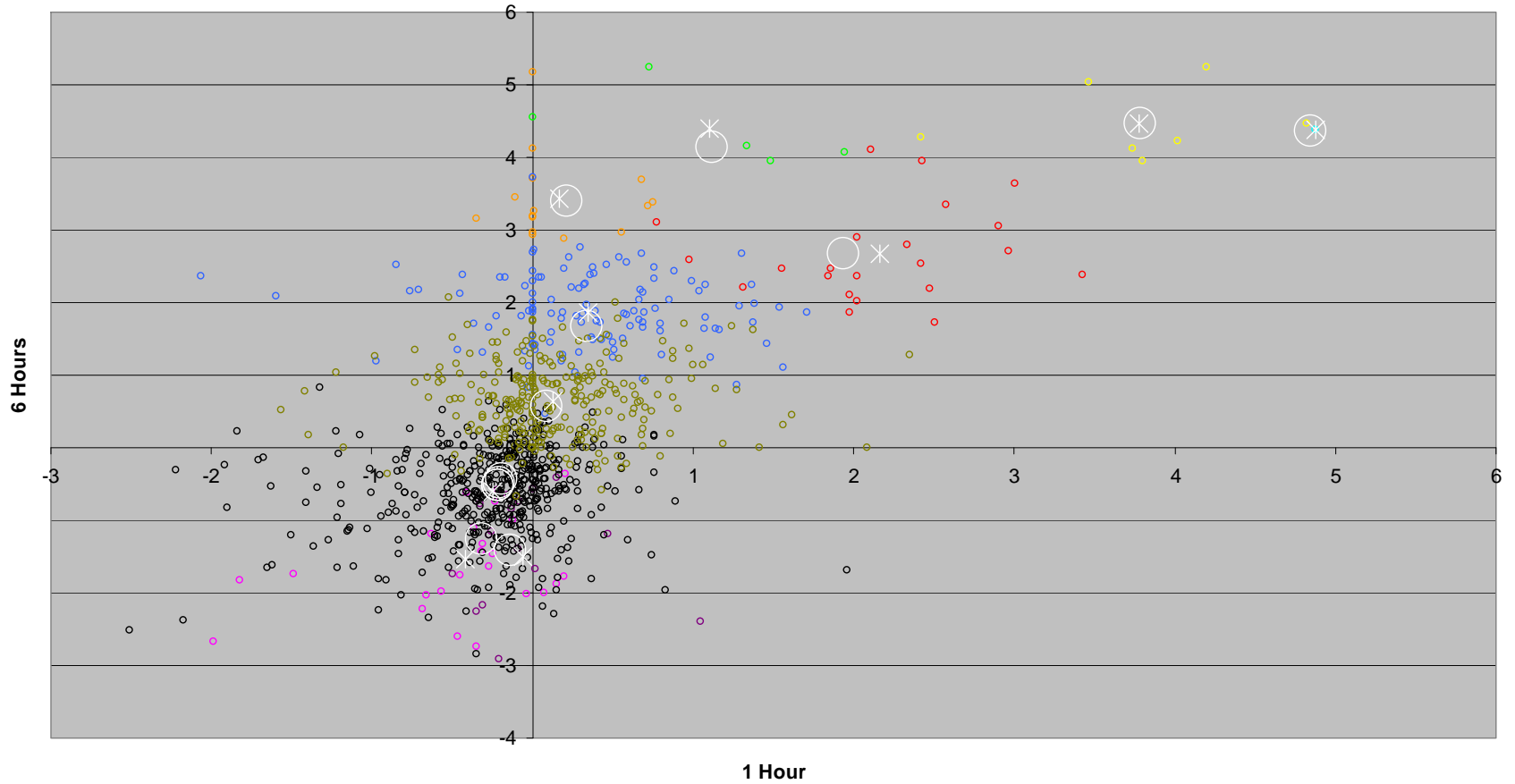
Ecoli

Cluster 0 Cluster 1 Cluster 2 Cluster 3 Cluster 4 Cluster 5 Cluster 6 Data Cluster 7 Rational Clusters Discrete Clusters



S Typhirium

○ Cluster 0 ○ Cluster 1 ○ Cluster 2 ○ Cluster 3 ○ Cluster 4 ○ Cluster 5 ○ Cluster 6 ○ Cluster 7 ○ Cluster 10 ○ Cluster 12 Rational Clusters Discrete Clusters



Self-Organizing Maps

- SOMs are based off of neural networks. They use a learning algorithm to train a random map to be like the input values. Thus similar inputs will be mapped close to each other

$n \leftarrow$ number of iterations for training algorithm

$V \leftarrow$ set of learning vectors (same dimension as input)

learning_algorithm(n, V)

 SOM \leftarrow random values

 for $j \leftarrow 1$ to n

$v \leftarrow V(\text{random})$

 BMU $\leftarrow \min(d(v, \text{SOM}))$ \\ for every node in SOM

 neighbors \leftarrow neighborhood(BMU)

 neighbors.weight = neighbors.weight + adjustment

end

- Kohonen, T., *Self-Organizing Maps*, Springer, Berlin, 1997.
- Torkkola, K., Gardner, R.M., Kayser-Kranich, T., and Ma, C., Self-organizing maps in mining gene expression data, *Information Sciences* 139 (2001) 79-96.
- Kohonen's Self organizing Feature Maps, *AI - Junkie*, March 11, 2009, <http://www.ai-junkie.com/ann/som/som1.html>.

Best Matching Unit

- The BMU is the minimum distance between the training vector and all the nodes in the SOM. It is typically found using the Euclidean distance formula.

$$d = \sqrt{\sum_{i=1}^k (v_i - w_i)^2}$$

Where k is the length of the input vectors, v is the training vector, and w is the weight vector from the current node in the SOM

- Kohonen, T., *Self-Organizing Maps*, Springer, Berlin, 1997.
- Torkkola, K., Gardner, R.M., Kayser-Kranich, T., and Ma, C., Self-organizing maps in mining gene expression data, *Information Sciences* 139 (2001) 79-96.
- Kohonen's Self organizing Feature Maps, *AI - Junkie*, March 11, 2009, <http://www.ai-junkie.com/ann/som/som1.html>.

Neighborhood

- The area of the neighborhood shrinks over time. You can use the exponential decay function for this.

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda}\right), \quad t = 1, 2, \dots, n$$

Where n is the number of iterations that the algorithm will run and σ_0 is the initial size of the neighborhood.

- Kohonen, T., *Self-Organizing Maps*, Springer, Berlin, 1997.
- Torkkola, K., Gardner, R.M., Kayser-Kranich, T., and Ma, C., Self-organizing maps in mining gene expression data, *Information Sciences* 139 (2001) 79-96.
- Kohonen's Self organizing Feature Maps, *AI - Junkie*, March 11, 2009, <http://www.ai-junkie.com/ann/som/som1.html>.

Weight Adjustment

- The weight is adjusted by multiplying two decay functions by the difference between the training vector and the SOM node.

$$w(t + 1) = w(t) + \theta(t)L(t)[v(t) - w(t)]$$

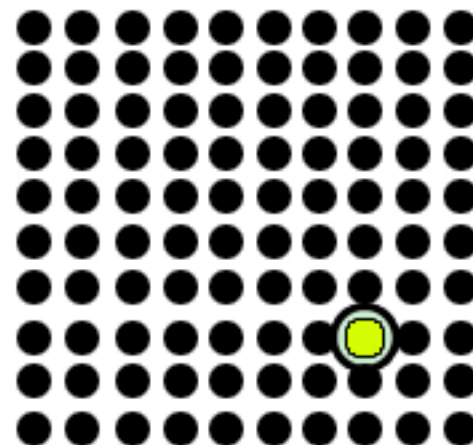
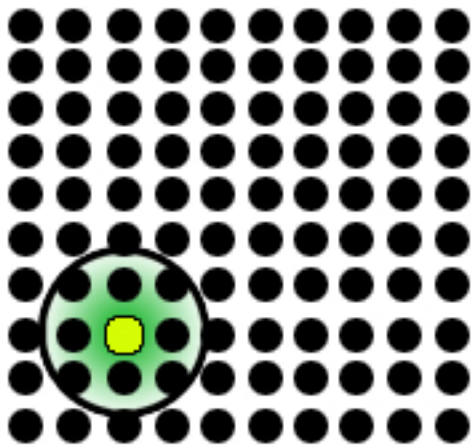
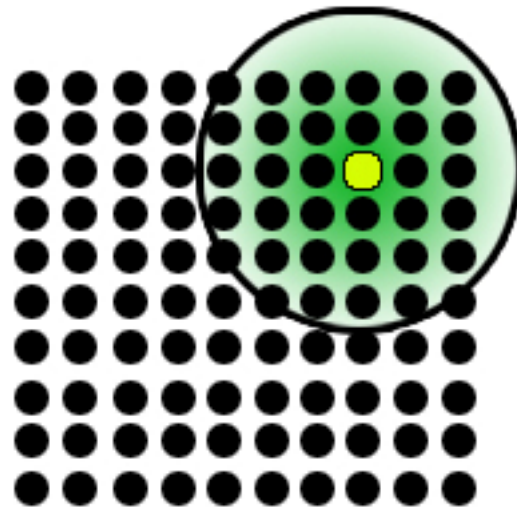
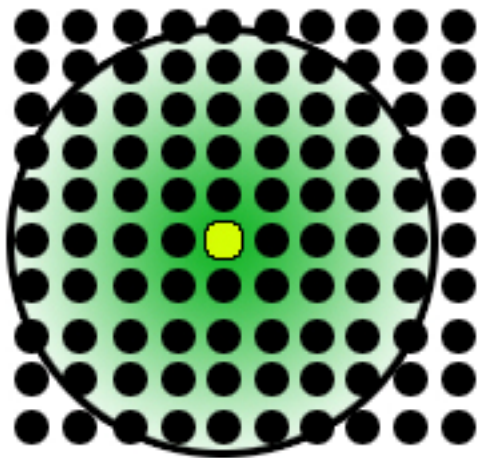
- One decay function decreases the learning variable by time.

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right) \quad t = 1, 2, \dots, n$$

- The other decay function decreases the rate of learning for neighbors further away from the BMU.

$$\theta(t) = \exp\left(-\frac{d^2}{2\sigma^2(t)}\right) \quad t = 1, 2, \dots, n$$

- Kohonen, T., *Self-Organizing Maps*, Springer, Berlin, 1997.
- Torkkola, K., Gardner, R.M., Kayser-Kranich, T., and Ma, C., Self-org
- Kohonen's Self organizing Feature Maps, *AI - Junkie*, March 11, 200



winning node

Mapping Mode (SOM)

- After the learning algorithm is ran on the map, the input is mapped onto the SOM and similar n-dimensional inputs will be near each other on the two-dimensional map

GenePattern

- <http://www.broad.mit.edu/cancer/software/genepattern/>
- **GenePattern** combines a powerful **scientific workflow platform** with more than **100 [genomic analysis tools](#)**.
- Use online or download

GenePattern

Module name: SOMClustering

Description: Self-Organizing Maps algorithm

Author: Keith Ohm (Broad Institute), gp-help@broad.mit.edu

Date: 10/28/03

Release: 1.0

Summary:

The Self Organizing Map (SOM) is a clustering algorithm where a grid of 2D nodes (clusters) is iteratively adjusted to reflect the global structure in the expression dataset. With the SOM, the geometry of the grid is randomly chosen (e.g., a 3 x 2 grid) and mapped to the k-dimensional gene expression space. The mapping is then iteratively adjusted to reflect the natural structure of the data. Resulting clusters are organized in a 2D grid where similar clusters lie near to each other and provide an automatic “executive” summary of the dataset. This module is a standard implementation of the SOM algorithm that can be used to cluster genes or samples (or just about any data, i.e. stocks, mutual funds, spectral peaks, etc).

Final Results

- <http://my.fit.edu/~sellings/finalClusters.zip>
- Arranged input data into 9 clusters using GenePattern software
- Why 9 clusters?
 - Used average of Stephen's output (8) + 1 to use a square map (3X3 SOM)

CLuster Identification via Connectivity Kernels (CLICK)

- Initialize Graph $G=(V,E)$
 - Vertex v : single gene “fingerprint” vector
 - Edge e : pairwise similarity between genes
 - Cut C : subset of E that partitions the graph
 - Cluster c : subset of V
 - Intersection of clusters $c_i, c_j, i \neq j$ is \emptyset
 - Fingerprint of a cluster c : *mean_vector(c)*

CLICK: Preprocessing

- Input: $n \times p$ matrix M of values
- n : Genes, p : Tests
- Data must be normalized
- Similarity measure:
 - $S_{vu} = v \cdot u = |v||u| \cos \theta$
 - Proportional only when norm is fixed for all $v \in V$

CLICK: Similarity

- Key idea: S is normalized, mixed distribution

For $u, v \in V$, mean = μ_T , variance = σ_T^2

$f(x | \mu_T, \sigma_T)$: pdf for elements in same cluster

For $u, v \in V$, mean = μ_F , variance = σ_F^2

$f(x | \mu_F, \sigma_F)$: pdf for elements in different clusters

Basic CLICK

- M : $n \times p$ matrix (genes vs. test conditions)
- S_{ij} : dot product of v_i, v_j
- w_{ij} : probability that v_i, v_j are *mates*

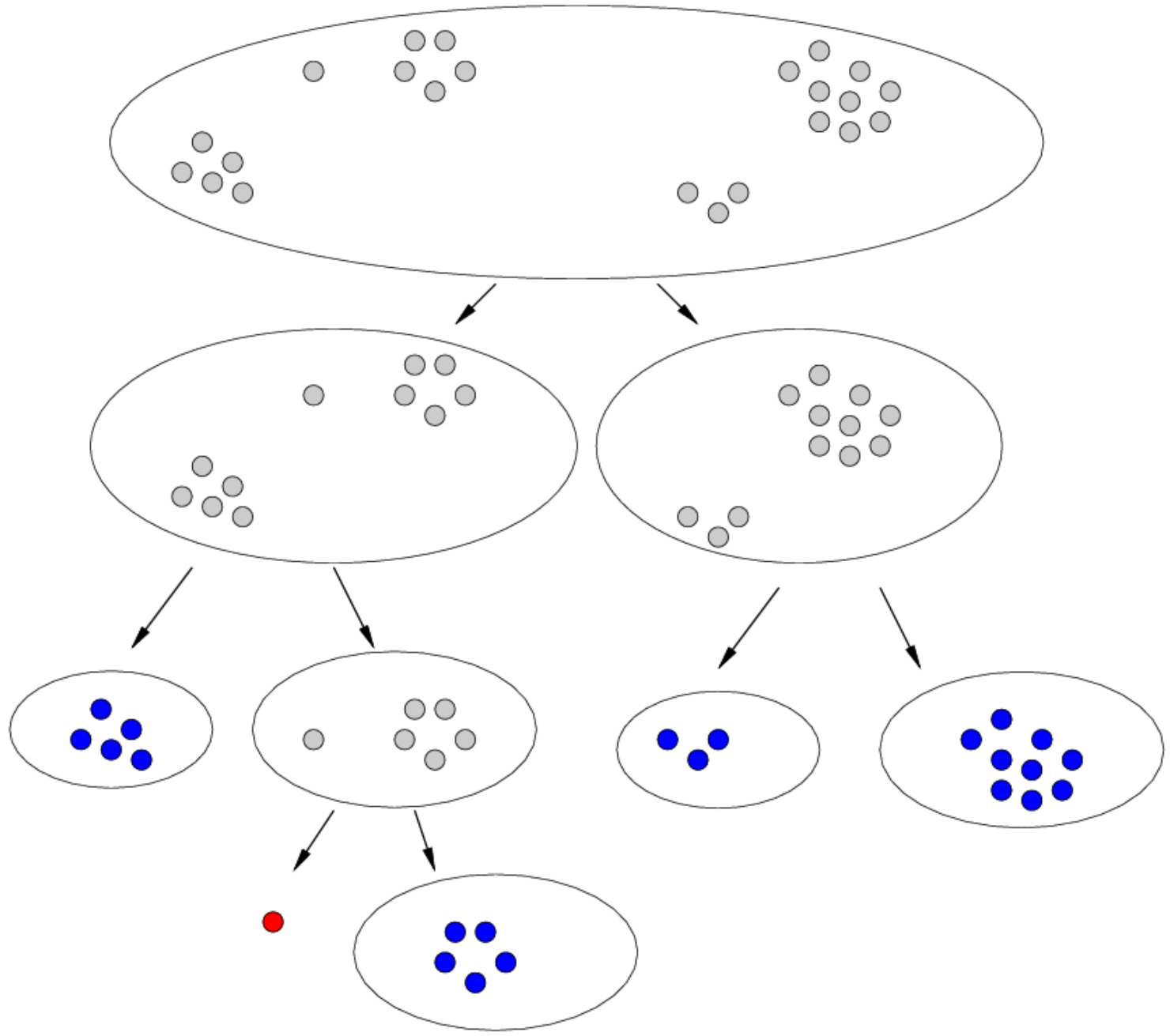
$$f(S_{ij} | \mu, \sigma) = (\sigma\sqrt{2\pi})^{-1} e^{-\frac{(S_{ij}-\mu)^2}{2\sigma^2}}$$

$$w_{ij} = \ln\left(\frac{p_{mates} f(S_{ij} | \mu_T, \sigma_T)}{(1 - p_{mates}) f(S_{ij} | \mu_F, \sigma_F)}\right)$$

$$w_{ij} = \ln\left(\frac{p_{mates} \sigma_F}{(1 - p_{mates}) \sigma_T}\right) + \frac{\sigma_T^2 (S_{ij} - \mu_F)^2 - \sigma_F^2 (S_{ij} - \mu_T)^2}{2\sigma_F^2 \sigma_T^2}$$

Basic CLICK

```
R: Singleton Set
Basic_CLICK(Graph G) {
  //v is a singleton
  If  $V(G) = \{v\}$  then R.add(v)
  Else if G is a kernel then
    Output(V(G))
  Else
    (H,K) <- MinWeightCut(G)
    Basic_CLICK(H)
    Basic_CLICK(K)
}
```



CLICK: Kernel

- Decision problem: is V ...
 - a singleton? ($|V| = 1$)
 - a subset of 2+ clusters? (need to partition more)
 - a subset of a single cluster? (kernel)

CLICK: Kernel

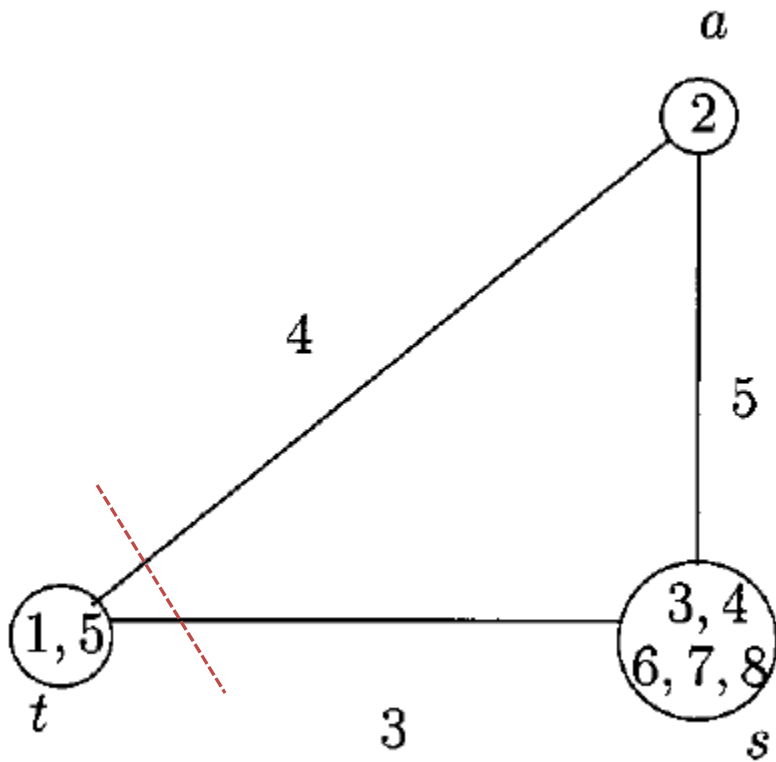
- For all possible cuts C connecting V :
 - H_0^C : Cut C disconnects two clusters
 - H_1^C : Cut C partitions a kernel
 - If $H_0^C > H_1^C$ for any C , then V is not a kernel
 - If V is not a kernel, then the graph should be partitioned into sub-graphs H, K

$$W(C) = \ln \left(\frac{\Pr(H_1^C | C)}{\Pr(H_0^C | C)} \right) = \sum_{(i,j) \in C} w_{i,j}$$

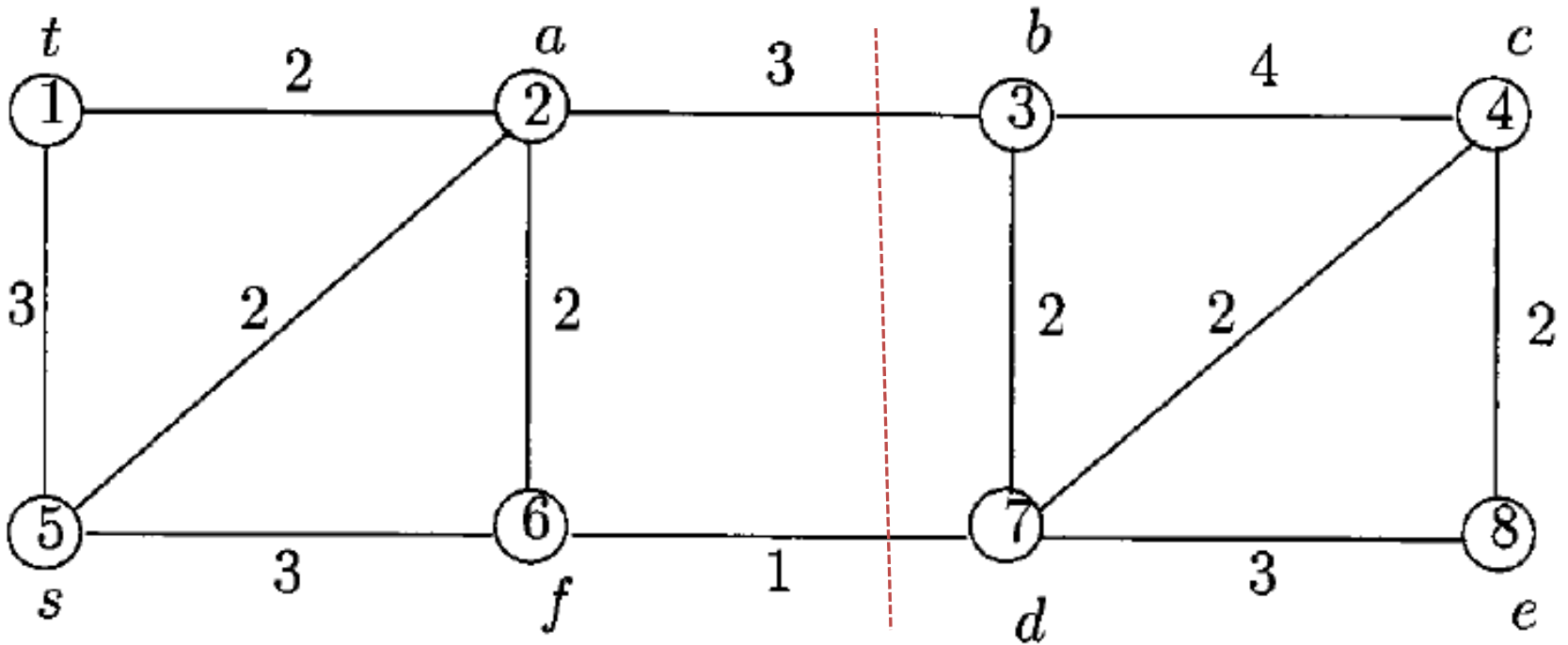
Minimal Weight Cut

- Choose one vertex as source, mark visited
- Mark each next highly connected vertex
- Last node, t , represents the cut:
 - $W(C_t) = \sum w_{i,t}$
 - Merge t with 2^{nd} last marked node, s
 - Remove t from V
 - Repeat until $|V| = 1$

MinWeightCut Example



MinWeightCut



CLICK: Adoption & Merging

- Basic_CLICK kernels – not full clusters
- Expand kernels by adding closest singletons
- Merge kernels with closest similarity
 - In both situations, only merge/adopt over some threshold

Full CLICK

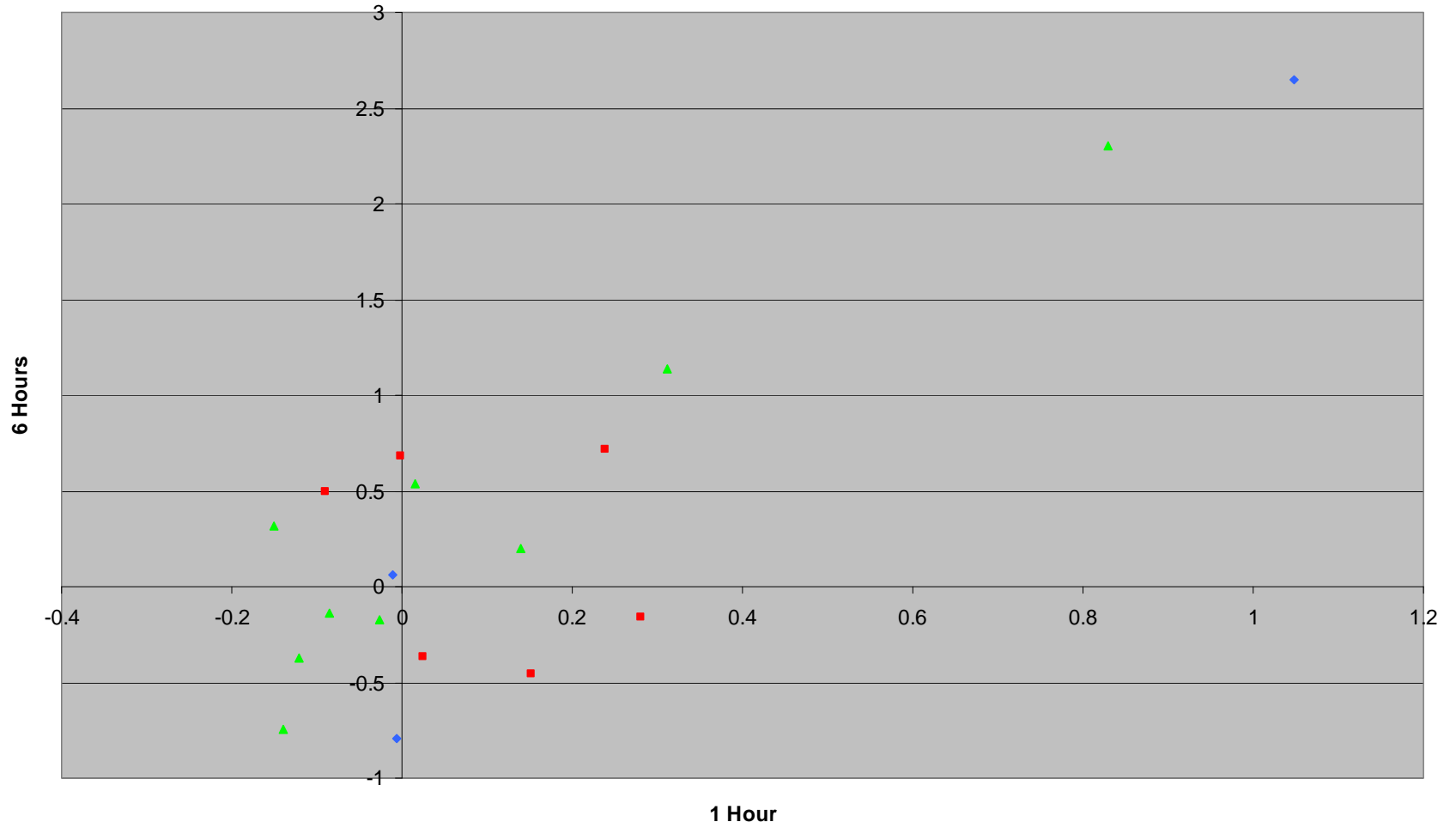
```
R: Singleton Set
Full_CLICK(Graph G = (V,E)) {
  R <- V
  While |R| is reduced {
    Basic_CLICK(G = (R,ER))
    Let L be the list of Kernels produced
    Let R be the set of Singletons produced
    Adopt(L, R)
  }
  Merge(L)
  Adopt(L, R)
}
```

Results & Analysis

- Two metrics:
 - Similarity between result clusters
 - Similarity of clusters with paper's results
- Map clusters to find regions of similar data
- Compare best clusters to find relationships

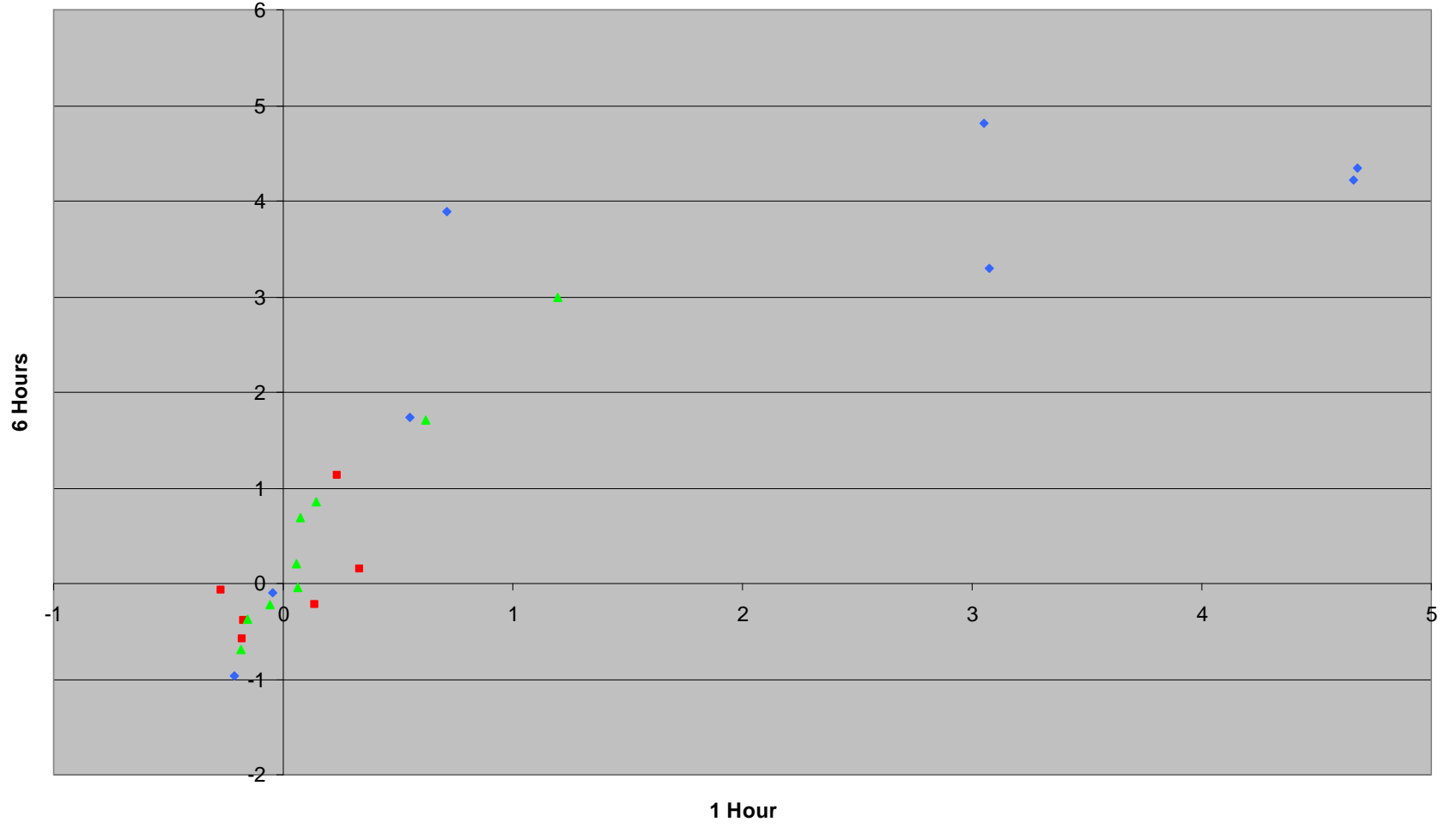
BCG

◆ Bayesian ■ Click ▲ SOM



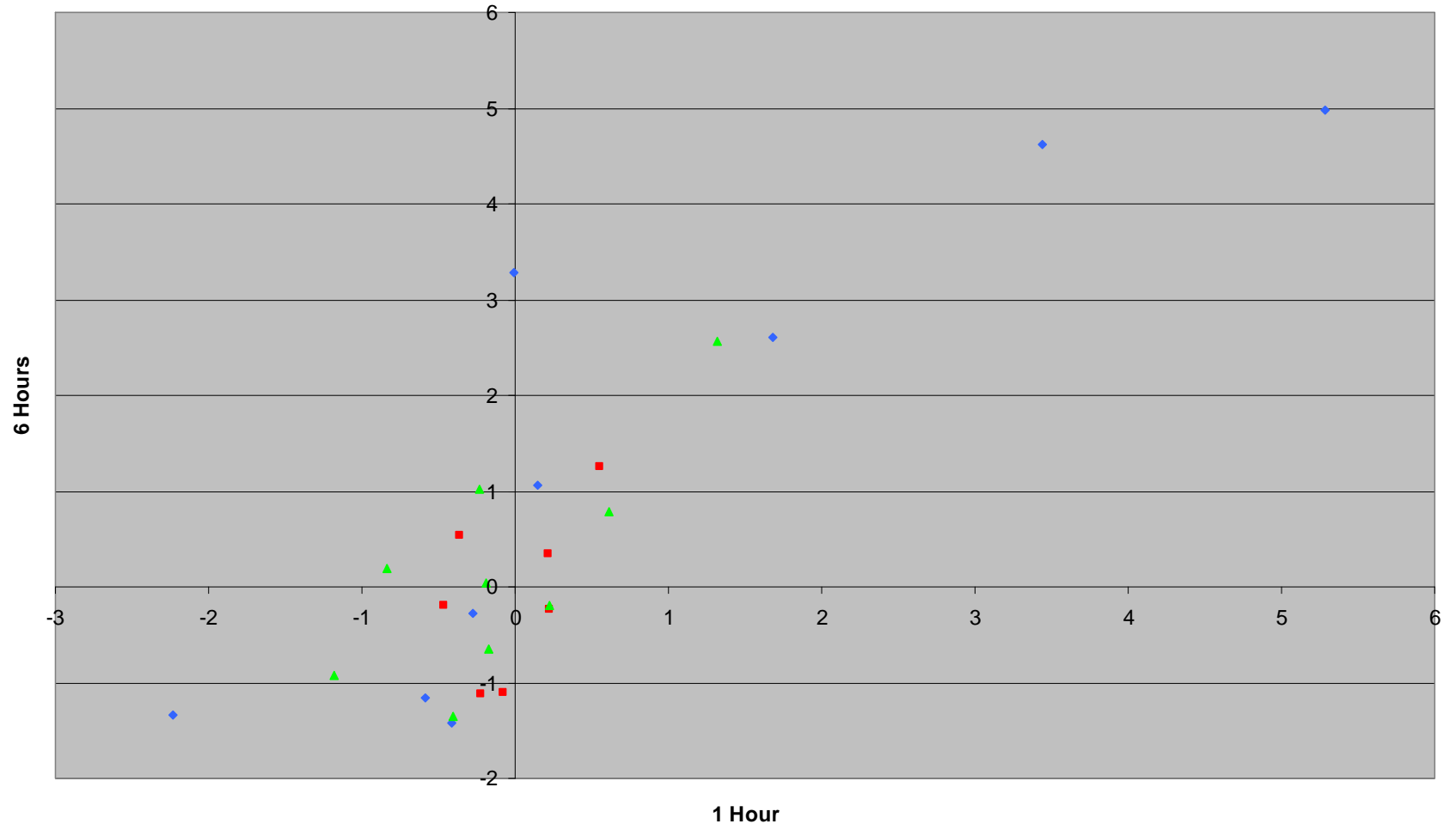
E. Coli

◆ Bayesian ■ Click ▲ SOM



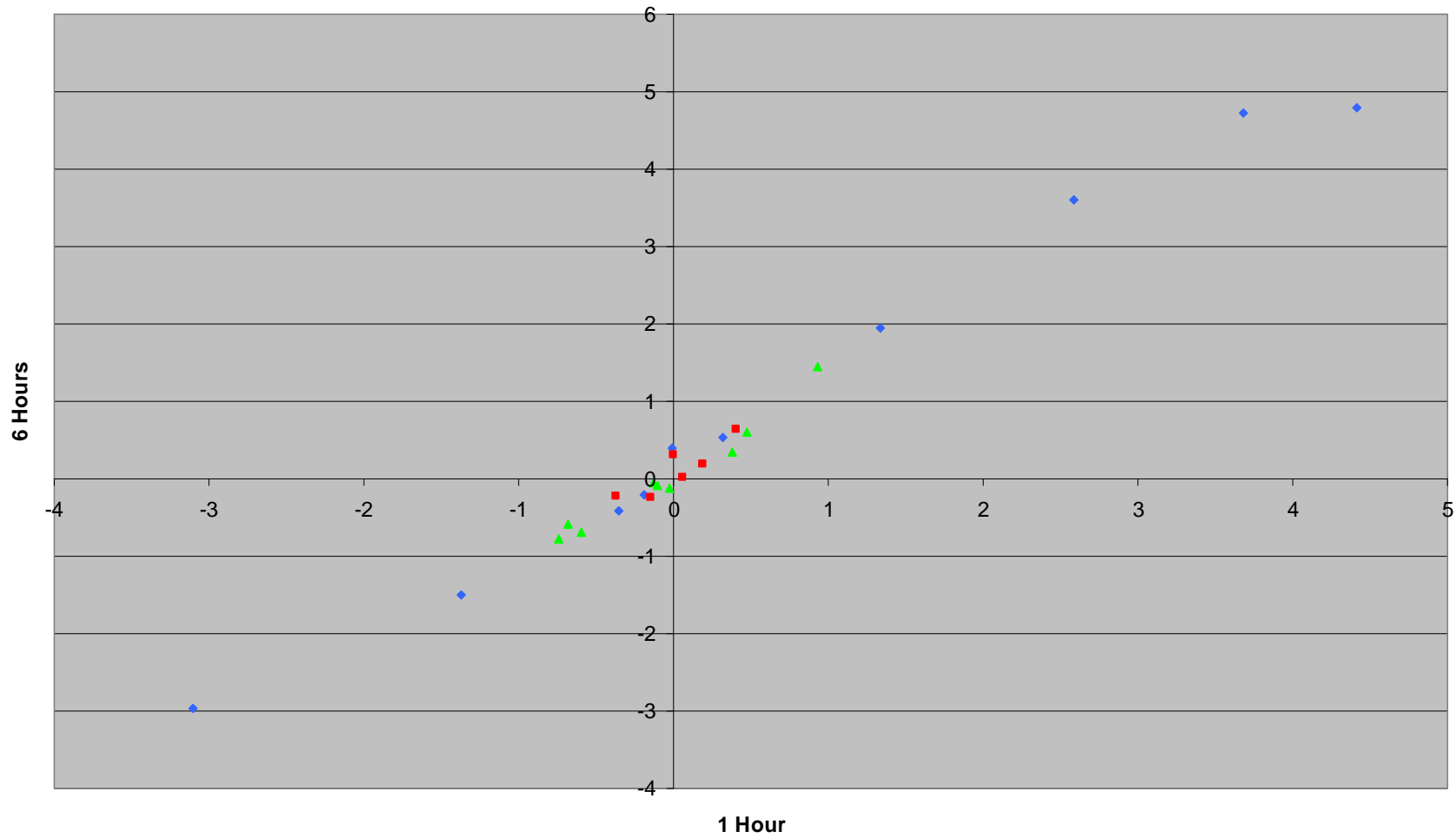
EHEC

◆ Bayesian ■ Click ▲ SOM



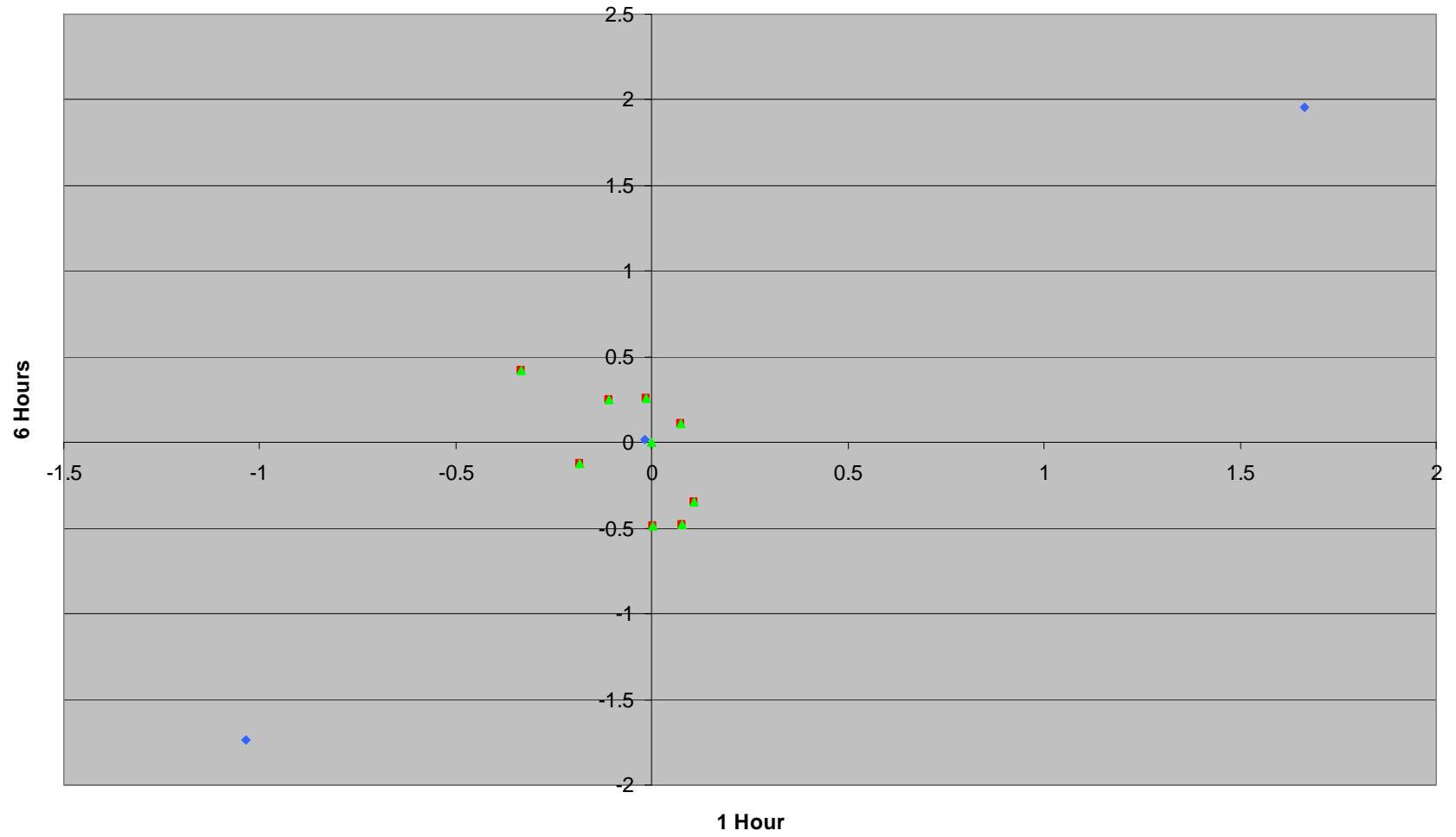
L Monocytogenes

◆ Bayesian ■ Click ▲ SOM



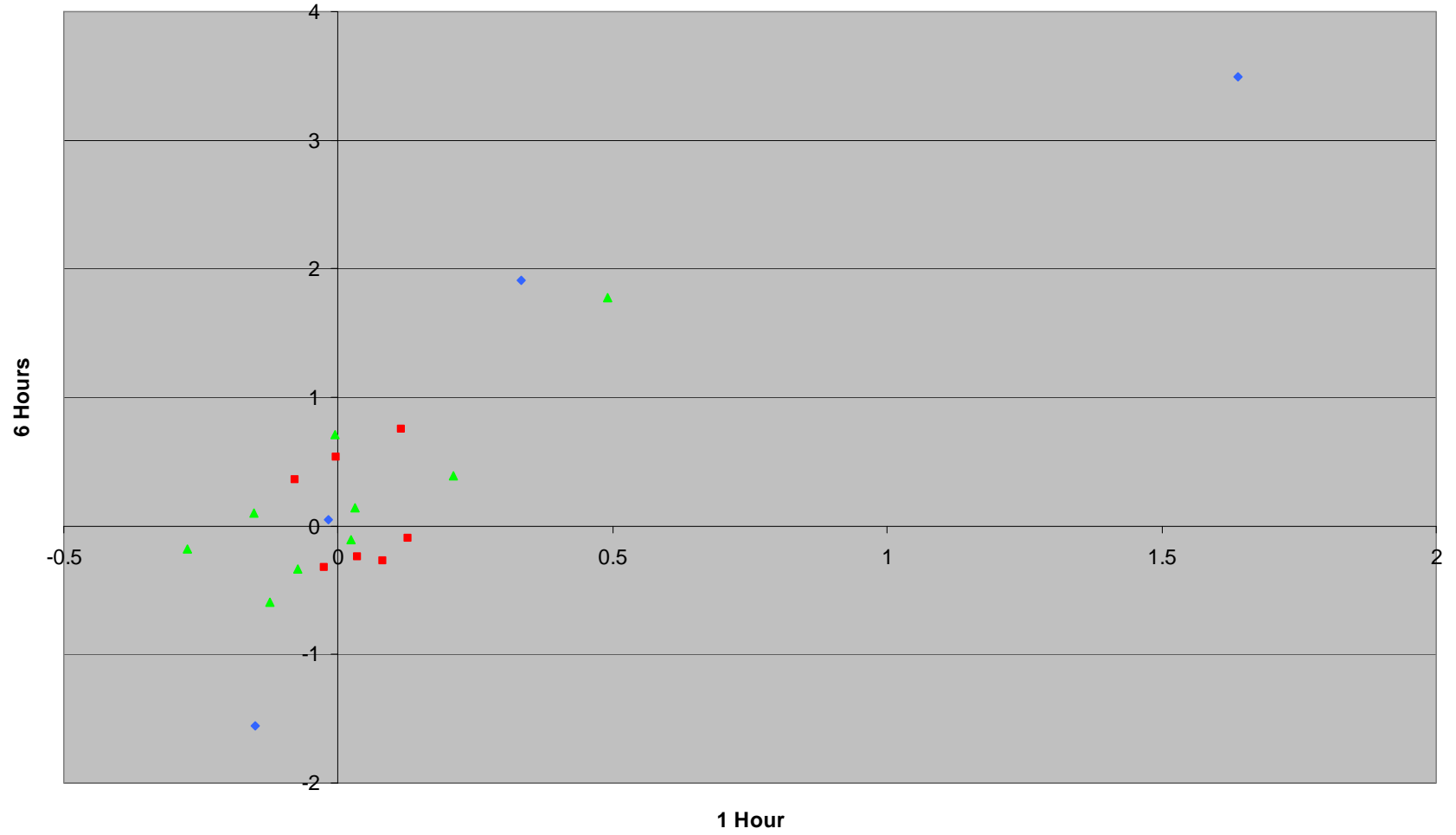
Latex

◆ Bayesian ■ Click ▲ SOM



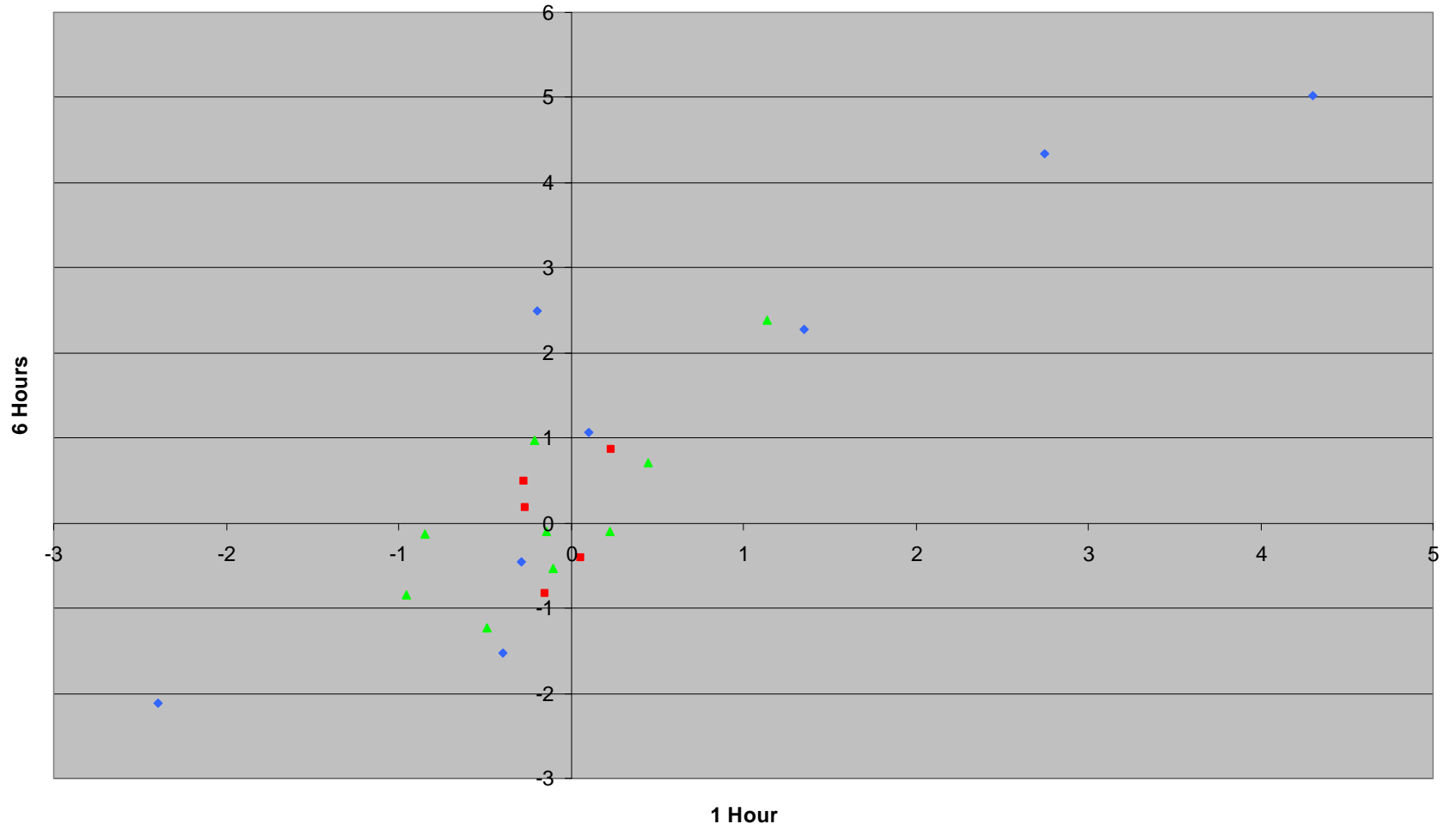
M Tuberculosis

◆ Bayesian ■ Click ▲ SOM



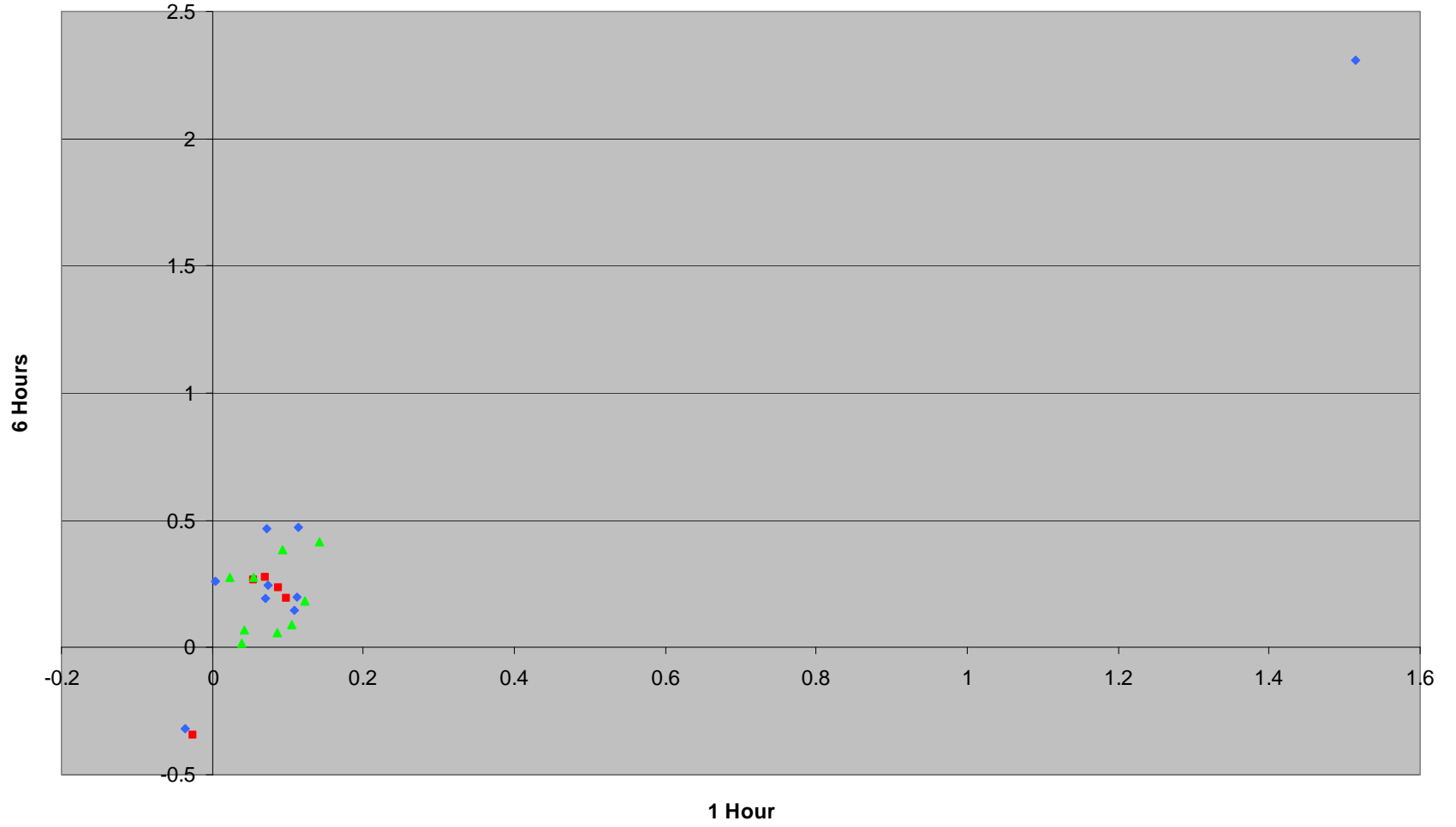
S Aureus

◆ Bayesian ■ Click ▲ SOM



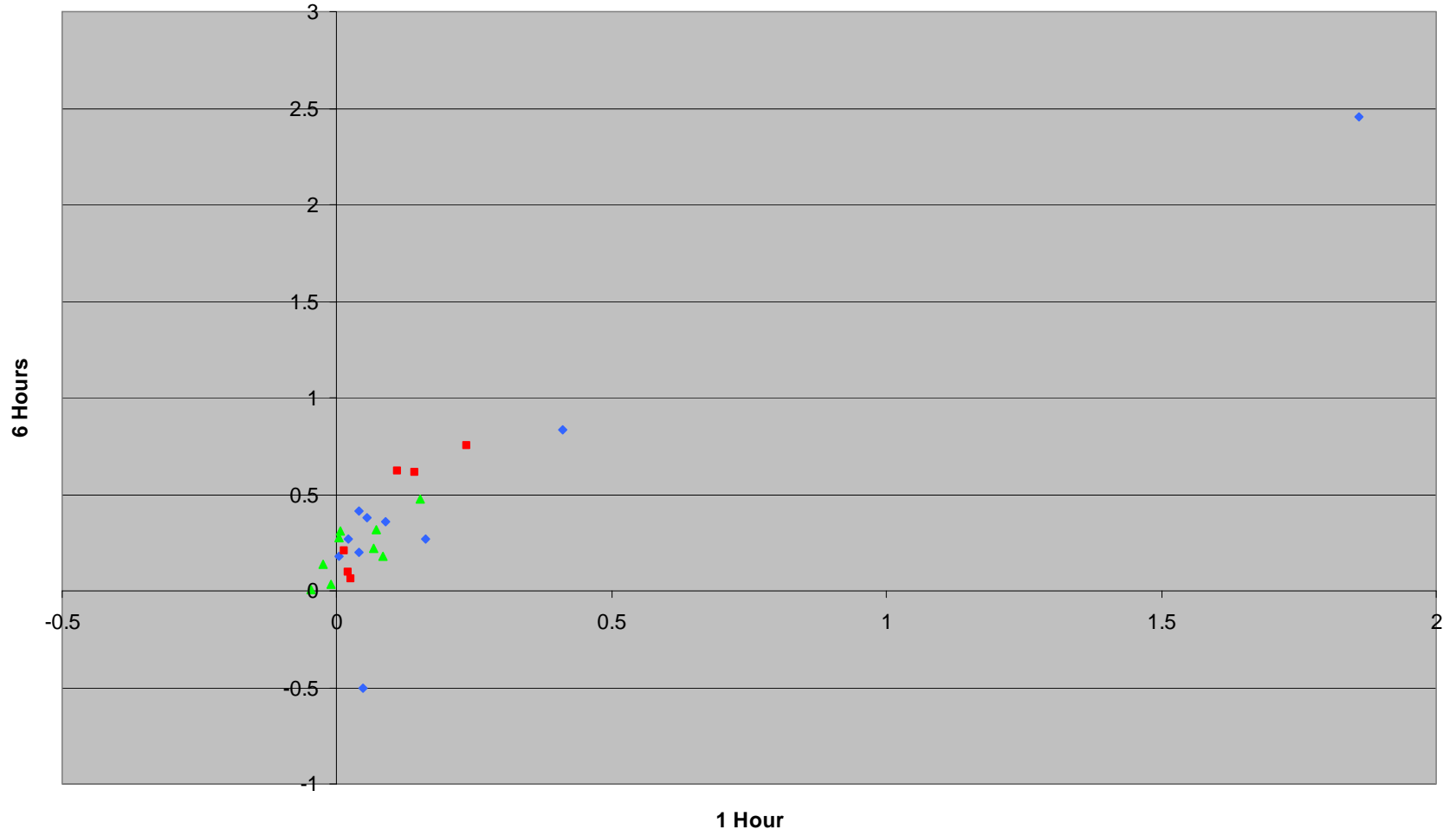
S Typhi

◆ Bayesian ■ Click ▲ SOM



S Typhimirium

◆ Bayesian ■ Click ▲ SOM



Error Metric

- Scores higher for clusters that either contain very many or very few clusters in the truth set.
- Scales to the number of datums in each cluster.
- Low error implies Truth cluster were highly correlated.
- High error implies Truth Clusters were well distributed.

$$E = \frac{\sum_{i=1}^k \left[\left(\frac{1}{2} - \left| \frac{1}{2} - \frac{|T \cap C_i|}{|T|} \right| \right) \cdot |C_i| \right]}{\sum_{i=1}^k |C_i|}$$

	Bayesian	Click	SOM
Error	4.33%	24.4%	18.22%

