Development of Complex Curricula for Molecular Bionics and Infobionics Programs within a consortial* framework**

Consortium leader

PETER PAZMANY CATHOLIC UNIVERSITY

Consortium members

SEMMELWEIS UNIVERSITY, DIALOG CAMPUS PUBLISHER

The Project has been realised with the support of the European Union and has been co-financed by the European Social Fund ***

**Molekuláris bionika és Infobionika Szakok tananyagának komplex fejlesztése konzorciumi keretben

***A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

INTRODUCTION TO BIOINFORMATICS

(BEVEZETÉS A BIOINFORMATIKÁBA)

CHAPTER 10

Gene Expression Analysis and Microarray Algorithms

(Gén kifejeződést elemző és mikróarray algoritmusok)

András Budinszky

Definitions

Gene expression is the process by which genetic information stored in a gene is “interpreted” and used in the synthesis of some functional gene product. Most often these products proteins, but in case of non-protein coding genes such as rRNA genes or tRNA genes, the product is a functional RNA.

The process of gene expression is used for generating the machinery for life and is the most fundamental level at which genotype gives rise to the phenotype through the properties of the expressed products.

Gene Expression Analysis

Observing the gene expression process and especially measuring the level of gene expression is very important for scientific work.

Quantifying the level at which a particular gene is expressed within a cell, tissue or organism is a significant information in many areas, such as:

- Identify viral infection of a cell
- Determine an individual's susceptibility to cancer
- Find if a bacterium is resistant to penicillin.
- Find genes with similar functionality

Gene Functionality

In a previous chapter we covered the process of gene finding. Once a gene is found we want to want to know the functionality of the newly discovered gene.

Simply comparing the new gene sequence to known DNA sequences (genes) often does not provide the function of gene.

For about 40% of sequenced genes, functionality cannot be determined by simply comparing to sequences of other known genes.

Measuring the expression level of the gen by the use of microarrays gives insight of the gene’s functionality even when sequence similarity alone is insufficient.

Two Major Uses for Measuring Gene Expression

A. Spotted array:

Typically, this is used to compare *two biological samples* (for example, healthy and cancerous tissue) because it applies two dyes. The color at the different spots indicates the expression level of the different genes.

B. Oligonucleotide array:

This is used for a *single biological sample* because it applies one dye. It measures the intensity of the expression level in the different points of the array (representing different genes). To obtain the expression levels for differently treated sample, the test should be repeated with multiple arrays. The results are stored in an intensity matrix.

Intensity Matrix

- Rows correspond to genes
- Columns correspond to different conditions (or points of time or species, etc)
- Elements are the intensities of hybridization signals read from the microarray

	Cond. 1	Cond. 2	Cond. 3
Gene 1	11	2	9
Gene 2	5	9	3
Gene 3	10	10	9
Gene 4	1	6	4
Gene5	4	8	3

Computational tasks in Microarray Analysis

A. Filtering the data

Obtain informative statistics for each data set intra-array and inter-array across all genes and detect anomalies.

Based on this observations, eliminate the “trouble-some” genes from the data.

B. Normalization of data

Transform the data set to make them comparable to one another in a probabilistic and statistical sense.

C. Interpretation of data

Analyze the corrected (filtered and normalized) intensity matrix to provide biological explanations.

Normalization of Data

The need for adjustment of the experimental data typically revealed by control experiments.

There three main normalization methods in common use:

A. Linear regression

B. Probabilistic distribution

Works under the assumption that the distribution of expression levels for duplicate sets of experiments should be identical.

C. Housekeeping genes and spike targets

Works under the premise that certain probes have a know constant behavior throughout all experimental conditions.

Interpretation of Data

After obtaining reliable intensities or intensity ratios and their variance of the experimental data, different data mining procedures can be applied.

The purpose of the gene expression experiment can influence the chosen data mining approach:

- Determining for an unknown gene the biochemical pathway it functions in
- Identifying group of genes that are co-regulated (and possibly function in the same pathway)
- Classifying specimens (e.g. tumors) according to their gene expression.

Distance of Instances (Genes)

For the purpose of the data mining process, the intensity values for each item (gene) in the matrix are considered coordinates of a point belonging to the item in a multi-dimensional space.

Thus, the major metric for the data mining is the distance among these points.

The distance measure should be symmetrical, and satisfy the triangle equality. It should be greater than 0 between two different points and if it is 0 than the two points must be identical.

Distance of Instances (cont)

Different distance measures can be used:

a) Euclidean distance (this one is used in most of the cases)

b) Pearson correlation coefficient – sensitive to outliers, especially when the number of points is small

c) Spearman correlation coefficient – like Pearson's except it converts the measurements to new values simply by sorting the old values and then them to their ranks (as 1, 2, 3, etc)

d) Absolute value of the correlation coefficients

Major Categories of Data Mining

Based on the fact whether or not outside information (that is, info not gained through this experiment) is used, the following two categories exist:

A. Supervised methods

Trains the data mining process with prior knowledge of classes of features.

For example, genes with known functionality and their expression patterns used as training set.

- Classification applies this.

B. Unsupervised methods

Only the data collected in the experiment is used.

- Clustering and principal component analysis applies this.

Distance Matrix

For each of the clustering algorithms, the intensity matrix (size *n*x*m*, where *n* is the number of genes, and *m* is the number of different measured values for a gene) is converted to an *n*x*n* symmetric distance matrix.

An element d_{ij} of this matrix is the computed distance between genes *i* and *j*.

For the distance computation any one of the distance functions mentioned on the previous slide can be used.

Distance Matrix, Example

	Cond. 1	Cond. 2	Cond. 3
Gene 1	11	2	9
Gene 2	5	9	3
Gene 3	10	10	9
Gene 4	1	6	4
Gene5	4	8	3

intensity matrix

	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
Gene 1	0	11	8.06	11.87	11
Gene 2		0	7.87	5.1	1.41
Gene 3			0	11.05	8.72
Gene 4				0	3.74
Gene 5					0

distance matrix (based on Euclidean distance)

Gene2 and Gene5 are the most similar ones

General Requirements for Clusters

Clusters should be created such a way that they satisfy the following two requirements:

A. Homogeneity

Elements within a cluster are close to each other.

This ensures high intra-cluster similarity.

B. Separation

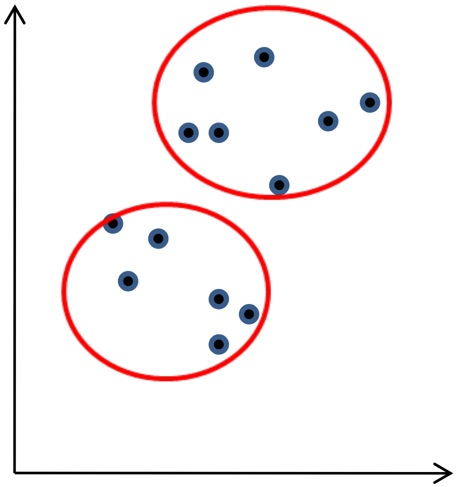
Elements in different clusters are further apart from each other.

This ensures low inter-cluster similarity.

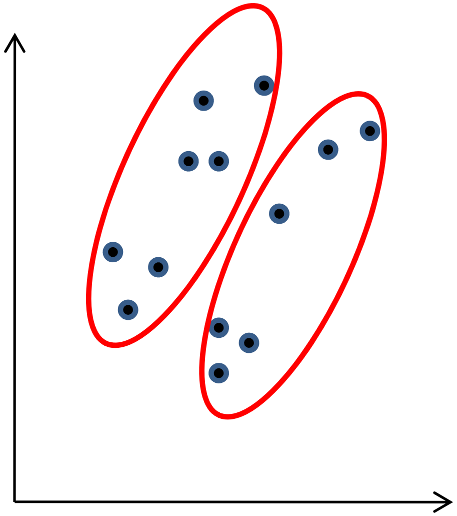
Note: This is not as easy to achieve as from the definition it sounds.

Homogeneity and Separation

Two clusterings of the same data points, satisfying and not satisfying the requirements



Good



Bad

Clustering

A. Hierarchical clustering

Groups the instances (genes) with small distance away together and forms a hierarchy (that is, a binary tree) of these groups.

Inheritedly creates embedded groups.

B. Partitional clustering

Uses also the distances of the instances in the decision making process

However, it groups the instances into mutually exclusive (disjoint) groups.

Hierarchical Clustering, Algorithms

A. Agglomerative (bottom-up) clustering

a) Start with *n* clusters, each containing one item (gene).

b) Find the two closest genes (or clusters in later iterations).

c) Merge them into one cluster.

d) Transform the distance matrix by deleting rows and columns of these two genes (clusters) and adding a row and column of the newly formed cluster with distances from the other genes/clusters computed (see distance computation between clusters on a follow-up slide).

e) Repeat steps b.) through d.) until arriving to a single cluster.

Hierarchical Clustering, Algorithms (cont)

B. Divisive (top-down) clustering

a) Start with a single cluster, containing all items.

b) Consider every possible way to divide the cluster (or any clusters in later iterations) and choose the best division (closest sub-clusters).

c) Execute the cluster splitting.

d) Repeat steps b.) through c.) until no more cluster to split.

Note: During the execution of either version of the algorithm, the constructed tree should be recorded.

Distance Computation of Clusters

Different strategies can be applied when computing the distance of two clusters:

A. Single linkage

   Distance is computed as the **smallest** distance between any pair of their elements helping to satisfy homogeneity.

B. Complete linkage

   Distance is computed as the **largest** distance between any pair of their elements helping to satisfy separation.

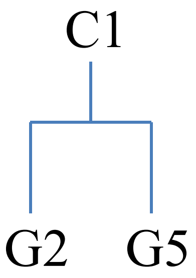
C. Average linkage

   Distance is computed as the **average** distance between any pair of their elements providing a balance homogeneity and separation.

Hierarchical Clustering, Example

	G1	G2	G3	G4	G5
G1	0	11	8.06	11.87	11
G2		0	7.87	5.1	1.41
G3			0	11.05	8.72
G4				0	3.74
G5					0

Gene2 and Gene5 are closest, therefore they form a cluster (C1)



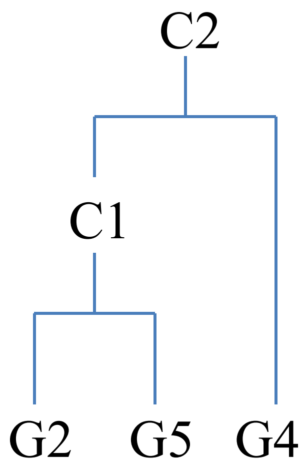
	G1	G3	G4	C1
G1	0	8.06	11.87	11
G3		0	11.05	8.30
G4			0	4.42
C1				0

Transformed distance matrix, C1 represents Cluster 1

Hierarchical Clustering, Example

	G1	G3	G4	C1
G1	0	8.06	11.87	11
G3		0	11.05	8.30
G4			0	4.42
C1				0

Gene4 and Cluster1 are closest, therefore they form next cluster (C2)



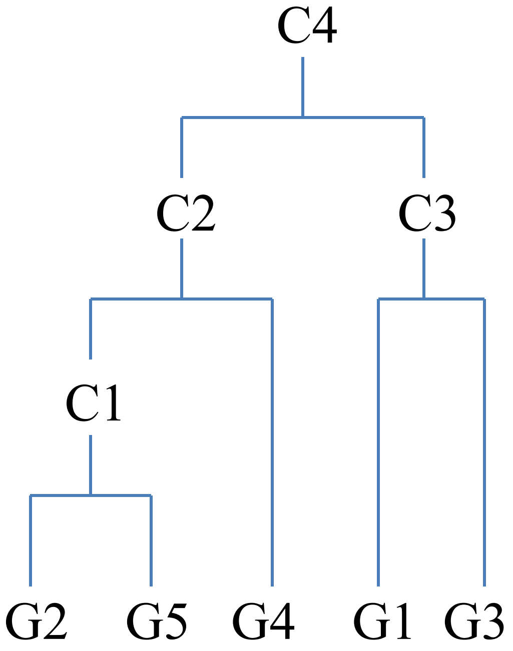
	G1	G3	G4
G1	0	8.06	11.29
G3		0	9.21
C2			0

Transformed distance matrix, C2 represents Cluster 2

Hierarchical Clustering, Example

	G1	G3	G4
G1	0	8.06	11.29
G3		0	9.21
C2			0

Gene1 and Gene3 are closest, therefore they form next cluster (C3), and then – as a last step – C2 and C3 combined into one cluster (C4)



Problems/Limitations of Hierarchical Clustering

1. Choice of distance function discussed earlier can influence the result.
2. No way of reassigning an incorrectly grouped item.
3. Choice of inter-cluster distance computation also affects the result. For example, single-linkage clustering often less successful than average-linkage clustering (joining clusters with distant centroids)

Note: It is a good strategy to try different choices to over-come problems mentions in 1. and 3.

k-Means Clustering

This is partitioning clustering method, and requires the prior specification of *k*, the number of clusters to be formed.

The distance matrix does not have to be generated because during the algorithm the distances are computed between data points and cluster centers (centroids).

The idea of the algorithm is that the items (genes) are assigned to clusters in such a way that the so-called squared error distortion is minimized:

$$d(\boldsymbol{V},\boldsymbol{X}) = \sum d(v_{i}, \boldsymbol{X})^{2} \ / \ n \qquad 1 \leq i \leq n$$

where $\boldsymbol{V}=\{v_1 \ldots v_n\}$ is the set of data points and
$\boldsymbol{X}$ is the set of *k* centroids

k-Means Clustering, Lloyd Algorithm

A. Choose *k*, the number of clusters, and one of the discussed distance functions to be used in each distance calculation during the rest of the algorithm.

B. Randomly partition the genes into *k* clusters.

C. Calculate the centroids of each cluster.

D. Reassign each gene to the cluster whose centroid is the closest.

E. Repeat steps C. and D. until no gene changes cluster membership.

Discussion of k-Means Lloyd Algorithm

The k-means clustering is an NP-hard problem.

The Lloyd algorithm is an efficient heuristic method which may lead just to a locally optimal clustering.

Even with heuristics it may require many iterations for any good size problem.

It is a good idea to execute the algorithm with different initial (seed) clustering because some choice may lead to bad result.

The algorithm also very sensitive to the value of *k*.

Choice of *k* Value

It is also a good idea to redo the process with different *k* values (2, 3, 4, etc).

A good way to decide how far to proceed when choosing *k* values is to compute the squared error distortion for each produced clustering.

Clearly when *k* increases then the squared error distortion value decreases.

Typically the proper *k* value can be picked by checking the plot of distortion as function of *k* and choosing a value where we get only a relatively small decrease.

Another (Conservative) k-Means Algorithm

In each iteration the Lloyd algorithm moves a lot of points (genes) from cluster to cluster.

The following k-means algorithm (conservatively) moves only one point in each iteration and only if the move improves the clustering (that is, decreasing the squared error distortion).

Steps:

A. Choose *k*, the number of clusters, and one of the discussed distance functions to be used in each distance calculation during the rest of the algorithm.

B. Randomly partition the genes into *k* clusters.

Conservative k-Means Algorithm

Steps (cont):

C. Check for each point (gene) how much it would lower (if at all) the squared error distortion in case this point is moved to any of the clusters it is not a member of.

D. Select the point and the cluster to cluster move that provides the best improvement in step C and execute that move.

E. If step C. finds no candidate for the move (that is, none of the moves would improve the clustering), then finish else repeat steps C. through E.

Self-Organizing Maps (SOM)

It is a multivariate data mining tool similar to k-means clustering.

It is a type of artificial neural network that is using a low-dimensional (typically two-dimensional or three-dimensional) grid of the nodes, called a map.

Then the data is mapped iteratively to the nearest node, one data point at a time until either fixed endpoints are reached or no more movement of nodes above a threshold.

Principal Component Analysis (PCA)

As it has been discussed, when applying clustering during gene expression analysis, we are looking for pattern among rows representing genes.

It is often the case that are correlations between some of the columns (variables: expression levels in the different experiments/microarrays).

The PCA determines if such sets of variables ($\{x_i\}$) exist and then selects new sets of variables $\{y_i\}$ that are uncorrelated.

In effect, it transforms the problem into a smaller-dimensioned space.

Graph-Based Partitioning Methods

These methods also produce partitioned clusters, but – as opposed to the previous methods – do not require the number of clusters to be created predefined.

Instead, they require a so-called affinity threshold (θ) parameter that specifies the minimum similarity necessary between an item (gene) and a cluster for becoming a member of that cluster.

Of course, the value of θ affects the number of clusters produced.

CAST(Cluster Affinity Search Technique)

This algorithm takes a graph-based approach that relies on the concept of a clique graph.

A clique graph is an undirected graph that is the union of disjoint complete graphs.

The distance matrix can be transformed into a distance graph where

- the vertices are the genes, and
- an edge is drawn between genes *i* and *j* if the distance between them is below the specified affinity threshold

Then ideally cliques (subgraphs) determine the clusters of the genes with similar characteristics.

Resources for Gene Expression Analysis

Gene Ontology Consortium

Provides functional description of genes. It has a standardized representation of gene and gene products across species and databases. Gene relationships are described by molecular function and biological processes, and represented in the form of DAGs

http://www.geneontology.org

DRAGON

It can be used for determining the relevant biological information for the many genes simultaneously expressed in a microarray experiment.

http://pevsnerlab.kennedykrieger.org/dragon.htm

Resources for Gene Expression Analysis (cont)

GenMapp

It is a free computer application designed to visualize gene expression and other genomic data on maps representing biological pathways and groupings of genes. Integrated are programs to perform a global analysis of gene expression.

http://www.genmapp.org/introduction.html

GoMiner

It is a tool for biological interpretation of 'omic' data – including data from gene expression microarrays.

http://discover.nci.nih.gov/gominer/