



*RELATIONAL CLUSTERING  
FOR KNOWLEDGE DISCOVERY  
IN LIFE SCIENCES*

A dissertation presented  
by

ILARIA GIORDANI

Submitted to the University of Milano-Bicocca  
in partial fulfillment of the requirements for the degree of  
DOCTOR of PHILOSOPHY

October 2009

Department of Informatics, Systems and Communication

Advisors: Prof. Enza MESSINA  
Prof. Francesco ARCHETTI

*To My Mother and My Father*

## Acknowledgments

First and foremost, I would like to thank my advisors Prof. Enza Messina and Prof. Francesco Archetti for their guidance, encouragement and patience. They help me in these years and in particular, encourage my scientific interests. It was a real pleasure working with them.

I am also grateful to Leonardo Vanneschi for being a guide and a friend, and for making the time spent together for our researches so enjoyable.

I am particularly grateful to all my colleagues and, between all, Elisabetta, Daniele and Cristina for their friendship, patience, understanding and for all the funny moments spent together.

I am also grateful to Consorzio Milano Ricerche for the opportunity of improving my skills. Thanks to Marco, Luigi and Federica that help me during this last hard months.

I would like to show my gratitude to Prof. Daniela Mari for her help and kindness during our works about the clinical study presented in this thesis. Thanks also to Giulia and Alessandra.

I would also like to express acknowledgement to all my friends, particularly to Francesco and Paolo that help me during my moments of sadness with hearty laughs. Thanks also to the “friends of the train”: with them I spent a lot of nice moments in these years.

Finally, I would thank my family, for their continual support, encouragement and for giving me the freedom to pursue my own interests. I cannot be grateful enough for their untiring support and unconditional belief in me.

Last but not the least, I could not have done any of this without Giacomo that gave me hope whenever I was down, strength when I felt weak and unquestioning love all along.

# Summary

Introduction.....	9
1. Clustering Analysis.....	11
1.1 Distance Measures .....	12
1.2 Clustering as an Optimization Problem.....	16
1.3 Clustering algorithms .....	17
1.3.1 Partitional K-Means Clustering Algorithm .....	17
1.3.2 Hierarchical Clustering Algorithm.....	21
1.4 Different Measures of Cluster Validation.....	23
1.4.1 Cluster Validity Measures for Partitional Clustering .....	24
1.4.2 Cluster Validity Measures for Hierarchical Clustering .....	28
2. Specific Challenges Related to Biomedical Data .....	30
2.1 Feature Selection.....	30
2.1.1 Supervised Feature Selection .....	31
2.1.2 Unsupervised Feature Selection .....	33
2.1.3 Proposed approach: Genetic Programming for Feature Selection.....	34
2.2 Mixed Data Types .....	51
2.2.1 Overview of Existing Algorithm .....	51
2.2.2 Modified K-prototypes Algorithm .....	53
2.3 Knowledge “Integration” .....	56
2.3.1 Knowledge in Life Science Domain .....	56
2.3.2 Knowledge Integration in Clustering Procedure.....	59
3. “Structure Driven” Methods .....	62
3.1 Biclustering Algorithm .....	62
3.2 3- Clustering Algorithm .....	67
3.3 Quality Measures for Biclustering and 3-Clustering.....	68
3.3.1 Quality Measure for Biclustering.....	68
3.3.2 Quality Measure for 3-Clustering .....	69
4. “Knowledge Driven” Methods.....	71
4.1 Constraint - Based Clustering Methods.....	71
4.2 Distance - Based Clustering Methods.....	72
4.3 Hybrid Methods: Combination of Distance-Based and Constraint-Based .....	73
4.4 The Proposed Relational Clustering Framework: Principal Features.....	74

5.	The Proposed Relational Clustering Framework: Case Studies.....	78
5.1	Learning Transcriptional Regulatory Modules .....	78
5.1.1	The biological problem: state of the art .....	78
5.1.2	Integration of Gene Regulatory Information and Gene Expression Data .....	80
5.1.3	The Proposed Iterative Relational Clustering Approach .....	81
5.1.4	Computational Experiments and Results .....	84
5.2	Detecting the Most Effective Cancer Drug: NCI-60 Dataset.....	90
5.2.1	The Pharmacogenomics Problem: State of the Art .....	90
5.2.2	Traditional Approaches: K-Means and SVT Algorithms.....	91
5.2.3	The Proposed Relational Clustering Approach .....	92
5.2.4	Computational Experiment and Results .....	94
5.3	Oral Anticoagulation Therapy .....	100
5.3.1	The Clinical Problem: State of the Art .....	100
5.3.2	Patient Profiling: Drug Sensitivity Index .....	102
5.3.3	Traditional Clustering Approach: a Modified Version of Mod-K-Prototype ..	106
5.3.4	The Proposed Relational Clustering Framework .....	109
5.3.5	Further Analysis on the Data Set .....	112
	Conclusion .....	118
	Appendix A: Data Resources .....	121
A.1	Transcription Factors Data .....	121
A.1.1	Gasch et al., 2000 Dataset.....	121
A.1.2	Spellman et al., 1998 Dataset.....	122
A.2	NCI-60 Data .....	125
A.3	Oral Anticoagulation Therapy (OAT) Data .....	127
A.4	Oncological Data .....	130
A.4.1	Colon Dataset .....	130
A.4.2	Leukemia Dataset.....	130
A.4.3	Molecular Dataset.....	130
	Bibliography.....	132

## List of Figures

Fig. 1-1: Typical cluster analysis procedure .....	11
Fig. 1-2: An example of better clustering done when using the Standardized Euclidean distance in comparison with the Euclidean distance. ....	14
Fig. 1-3: The Manhattan vs. Euclidean distance .....	14
Fig. 1-4: Expression levels in different sample of two genes.....	16
Fig. 1-5: Example of using the K-means algorithm to find three clusters in sample data .....	18
Fig. 1-6: A hierarchical clustering of four points shown as a dendrogram and as nested clusters .....	22
Fig. 1-7: Graph-based definitions of cluster proximity .....	23
Fig. 2-1: Taxonomy of feature selection techniques. ....	32
Fig. 2-2: An example of a simple GP individual.....	36
Fig. 2-3: Normalized Z-score of the most recurrent common genes for the Colon dataset.....	49
Fig. 2-4: Normalized Z-score of the most recurrent common genes for the Leukemia dataset. ....	50
Fig. 2-5: Schema of modk-prototypes (Bushel et al., 2007) .....	53
Fig. 2-6: Reductionist approach and Integrative approach. ....	56
Fig. 2-7: Iterative in silico model building in biology .....	57
Fig. 2-8: Beyond Genomics Correlation Network .....	58
Fig. 2-9: Top down versus bottom-up approaches. ....	59
Fig. 2-10: Modified clustering procedure with knowledge integration.....	60
Fig. 3-1: gene expression dataset .....	62
Fig. 3-2: Perfect constant bicluster.....	64
Fig. 3-3: Bicluster with constant values on rows or columns .....	64
Fig. 3-4: Bicluster with coherent values .....	65
Fig. 3-5: Bicluster with coherent evolution .....	65
Fig. 3-6: Sample data and clusters.....	67
Fig. 4-1: Relational Clustering Core .....	75
Fig. 4-2: Relation Learning Phase.....	76
Fig. 5-1: QTF matrix .....	80
Fig. 5-2: Instantiation of general relational clustering framework.....	82
Fig. 5-3: Iterative process of learning distance measure and modify objective function.....	83
Fig. 5-4: Number of co-expressed and co-regulated clusters for (Gasch et al., 2000) dataset .....	86
Fig. 5-5: Number of co-expressed and co-regulated clusters for (Spellman et al., 1998) dataset .....	87
Fig. 5-6: Genes correctly predicted by iterative relational clustering approach for (Gasch et al., 2000) dataset. ....	88
Fig. 5-7: Genes correctly predicted by STVQ approach for (Gasch et al., 2000) dataset.....	88
Fig. 5-8: Genes correctly predicted by iterative relational clustering approach for (Spellman et al., 1998) dataset .....	88
Fig. 5-9: Genes correctly predicted by STVQ approach for (Gasch et al., 2000) dataset.....	88
Fig. 5-10: General “flat” representation of cell lines .....	92
Fig. 5-11: Instantiation of the general relational clustering framework .....	92
Fig. 5-12: Computational process of the proposed relational clustering framework.....	94
Fig. 5-13: A sub-sample of the obtained clustering solution and a particular sub-pattern representing an example of the active drugs.....	97
Fig. 5-14: Bayesian Networks for modelling the NCI60 dataset .....	98
Fig. 5-15: Therapeutic INR range .....	101
Fig. 5-16: Oral Anticoagulation Therapy workflow.....	101
Fig. 5-17: Drug Sensitivity distribution .....	103
Fig. 5-18: Approximate Entropy distribution for entire dataset.....	105

Fig. 5-19: Approximate entropy for each $D_{sens}$ class .....	105
Fig. 5-20: Approximate Entropy for patients with different age .....	106
Fig. 5-21: schema of the OAT Mod-k-prototypes algorithm.....	106
Fig. 5-22: $\alpha$ and $\beta$ parameter .....	107
Fig. 5-23: DVI_CU index variation for k from 0 to 20 for dataset $\Omega_1$ .....	108
Fig. 5-24: DVI_CU index variation for k from 0 to 20 for genomic dataset $\Omega_2$ .....	108
Fig. 5-25: Instantiation of general relational clustering framework.....	109
Fig. 5-26: Computational process of the proposed relational clustering framework.....	111
Fig. 5-27: Wild type patient, positive Drug Sensitivity class .....	116
Fig. 5-28: Patient with two polymorphisms (gene CYP2C9: AC; gene VKORC1: TT), medium Drug Sensitivity class .....	116
Fig. 5-29: Patient with two polymorphisms (gene CYP2C9: CC; gene VKORC1: CT), negative Drug Sensitivity class .....	117
Fig. A - 1: (Gasch et al., 2000) dataset representation .....	122
Fig. A - 2: (Spellman et al., 1998) data set representation .....	123
Fig. A - 3: Simplified schematic overview of NCI60 database.....	125
Fig. A - 4: dendrogram showing average-linkage hierarchical clustering of human cancer cell lines .	126
Fig. A - 5: Relational model of OAT application .....	127
Fig. A - 6: CYP2C9 genotypes prevalence and the mean weekly maintenance dosing for Warfarin....	129
Fig. A - 7: VKORC1 genotypes prevalence and the mean weekly maintenance dosing for Warfarin...	129

## List of Tables

Tab. 1-1: Ideal cluster similarity matrix .....	26
Tab. 1-2: Ideal classification similarity matrix .....	26
Tab. 1-3: Two-way contingency table for determining .....	26
Tab. 2-1: Parameters used in the presented GP experiments.....	36
Tab. 2-2: Experimental results returned by Linear Regression for therapeutic responses prediction of four different drugs.....	38
Tab. 2-3: Experimental results returned by Least Square Regression for therapeutic responses of four different drugs. ....	39
Tab. 2-4: Results returned by GP .....	40
Tab. 2-5: Results that we have obtained performing 100 independent runs of RMSEGP on our dataset. ....	42
Tab. 2-6: Experimental results returned by Linear Regression .....	43
Tab. 2-7: Experimental results returned by Least Square Regression.....	43
Tab. 2-8: Results obtained with CCGP configuration.....	44
Tab. 2-9: Results obtained with linScalGP configuration.....	44
Tab. 2-10: Results returned by non- evolutionary methods on Colon dataset .....	46
Tab. 2-11: Results returned by the studied GP variants on the Colon dataset .....	46
Tab. 2-12: Results returned by the non evolutionary methods on the Leukemia dataset.....	47
Tab. 2-13: Results returned by the studied GP variants on the Leukemia dataset .....	47
Tab. 5-1: Iterative relational clustering algorithm results on (Gasch et al., 2000) dataset .....	86
Tab. 5-2: Iterative relational clustering algorithm results on (Spellman et al., 1998) dataset .....	86
Tab. 5-3: Transcription factors found by our iterative relational clustering algorithm on (Gasch et al., 2000) dataset .....	88
Tab. 5-4: Transcription factors found by our iterative relational clustering algorithm applied on (Spellman et al., 1998) dataset .....	89
Tab. 5-5: Computational results on $\Omega_1$ .....	96
Tab. 5-6: Computational results on $\Omega_2$ .....	96
Tab. 5-7: Computational results of Bayesian networks on $\Omega_2$ .....	99
Tab. 5-8: F-measure and entropy results for OAT modify k prototypes algorithm .....	108
Tab. 5-9: F-measure and entropy results for OAT modify k prototypes algorithm .....	112
Tab. 5-10: INR based classification results .....	113
Tab. 5-11: Drug sensitivity based classification results .....	113
Tab. 5-12: Drug sensitivity based classification results with new features .....	114
Tab. 5-13: Induction phase: Drug sensitivity based classification results .....	114
Tab. 5-14: INR based classification results on $\Omega_2$ dataset configuration .....	114
Tab. 5-15: Dsens based classification with genomic data results. In this phase INR average and variance are not considered. ....	115
Tab. 5-16: Dsens based classification with complete genomic data results .....	115
Tab. 5-17: Dsens based classification with genomic data results. In this phase INR average and variance are not considered. ....	115
Tab. 5-18: Genomic variant distribution in the three Dsens classes .....	116
Tab. A – 1: OAT patients’ characteristics .....	128
Tab. A - 2: Allelic variant frequencies of gene CYP2C9 and VKORC1 .....	128



# Introduction

Clustering is one of the most common machine learning technique, which has been widely applied in genomics, proteomics and more generally in Life Sciences.

In particular, clustering is an unsupervised technique that, based on geometric concepts like distance or similarity, partitions objects into groups, such that objects with similar characteristics are clustered together and dissimilar objects are in different clusters.

In many domains where clustering is applied, some background knowledge is available in different forms: labelled data (specifying the category to which an instance belongs); complementary information about “true” similarity between pairs of objects or about the relationships structure present in the input data; user preferences (for example specifying whether two instances should be in same or different clusters). In particular, in many real-world applications like biological data processing, social network analysis and text mining, data do not exist in isolation, but a rich structure of relationships subsists between them. A simple example can be viewed in biological domain, where there are a lot of relationships between genes and proteins based on many experimental conditions. Another example, maybe common, is the Web search domain where there are relations between documents and words in a text or web pages, search queries and web users.

Our research is focalized on how this background knowledge can be incorporated into traditional clustering algorithms to optimize the process of pattern discovery (clustering) between instances.

In this thesis, we first provide an overview of traditional clustering methods with some important distance measures and then we analyze three particular challenges that we try to overcome with different proposed methods: “feature selection” to reduce high dimensional input space and remove noise from data; “mixed data types” to handle in clustering procedure both numeric and categorical values, typically of life science applications; finally, “knowledge integration” in order to improve the semantic value of clustering incorporating the background knowledge.

Regarding the first challenge we propose a novel approach based on using of genetic programming, an evolutionary algorithm-based methodology, in order to automatically perform feature selection.

Different clustering algorithms are been investigated regarding the second challenge. A modify version of a particular algorithm is proposed and applied to clinical data.

Particularly attention is given to the final challenge, the most important objective of this Thesis: the development of a new relational clustering framework in order to improve the semantic value of clustering taking into account in the clustering algorithm relationships learned from background knowledge.

We investigate and classify existing clustering methods into two principal categories:

- *Structure driven approaches*: that are bound to data structure.

The data clustering problem is tackled from several dimensions: clustering

concurrently columns and rows of a given dataset, like biclustering algorithm presented in subsection 3.1 or vertical 3-D clustering presented in subsection 3.2.

- *Knowledge driven approaches*: where domain information is used to drive the clustering process and interpret its results: semi-supervised clustering (presented in subsection 4), that using both labelled and unlabeled data, has attracted significant attention. This kind of clustering algorithms represents the first step to implement the proposed general framework that it is classified into this category.

In particular the thesis focuses on the development of a general framework for relational clustering instantiating it for three different life science applications: the first one with the aim of finding groups of genes with similar behaviour respect to their expression and regulatory profiles. The second one is a pharmacogenomics application, in which the relational clustering framework is applied on a benchmark dataset (NCI60) to identify a drug treatment to a given cell line based both on drug activity pattern and gene expression profile. Finally, the proposed framework is applied on clinical data: a particular dataset containing different information about patients in anticoagulant therapy has been analyzed to find groups of patients with similar behaviour and responses to the therapy.

### **Thesis Outline**

This thesis is organized as follows: standard clustering techniques and distance measures will be outlined in section 1.

Three specific challenges related to life science domain will be illustrated into section 2. In particular, subsection 2.1 is focalized on feature selection, generally used to reduce high dimensional input space, and on the proposed technique based on Genetic Programming. In subsection 2.2 different clustering algorithm for mixed data type, a typical problem that born when we apply clustering algorithm to life science data, are investigated and in particular the attention is focalized on the approach called Modify-k-prototypes. Last subsection (2.3) is dedicated to the investigation of the different kind of information that we possibly integrate into the clustering process and on a general presentation of relational clustering.

A review of structure driven approaches and knowledge driven approach developments is covered respectively in sections 3 and 4.

In particular, in section 4 the proposed relational clustering framework that has been applied to three case studies is presented.

Section 5 described these case studies with, for each one, an introduction on the problem, the presentation of the instantiation of the general framework and, finally, the promising results obtained.

A description of data used in this Thesis is presented into appendix A.

# 1. Clustering Analysis

Human beings are skilled at dividing objects into groups (clustering) and assigning particular objects to these groups (classification).

Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships. The goal is that the objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct the clustering.

Cluster analysis is related to other machine learning techniques that are usually used to divide data objects into groups (subsets or categories), like classification.

In particular, classification is a form of *supervised learning* i.e., new unlabeled objects is assigned a class label using a model developed from objects with known class labels “the training set”. In contrast, cluster analysis that creates an implicit labelling of objects with class (cluster) labels, derived only from data, is referred to a form of *unsupervised learning* (no labelled data are available).

Typical cluster analysis consists of four principal steps (Xu and Wunsch, 2005) that are closely related each other and affect the derived clusters.

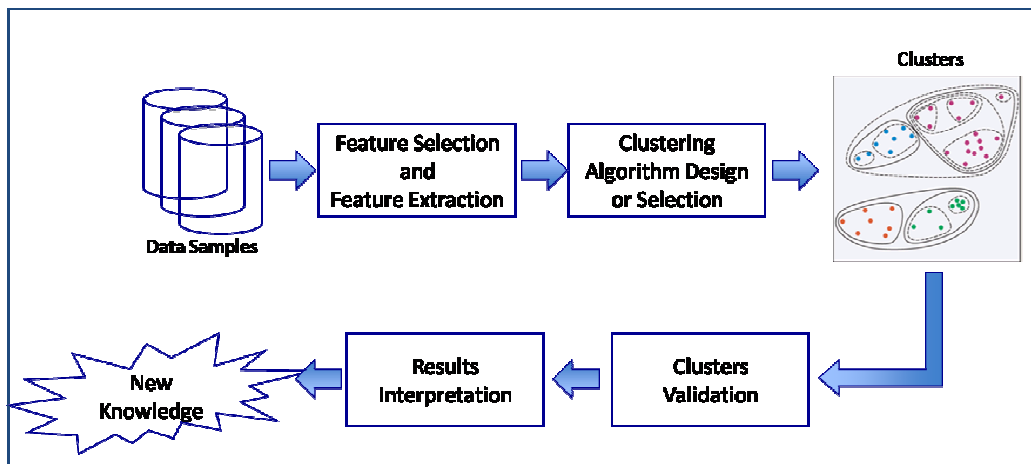


Fig. 1-1: Typical cluster analysis procedure

Fig. 1-1 depicts the procedure of traditional cluster analysis with four basic steps:

1) **Feature selection and extraction:** In general, as pointed out by (Jain et al., 2000) and (Bishop, 1995), *feature selection* chooses distinguishing features from a set of candidates, while *feature extraction* utilizes some transformations to generate useful and novel features from the original ones. Both are very crucial to the effectiveness of clustering applications: the selection of features can simplify the clustering process. Generally, ideal features should be of use in distinguishing patterns belonging to different clusters, immune to noise, easy to extract and interpret. In section 2.1 we will discuss on feature selection techniques.

2) **Clustering algorithm design or selection.** This step represents the most important phase of all clustering procedures. Here a similarity measure between objects must be selected and a criterion function, which minimize the similarity between objects belonging to the same cluster and maximize the similarity between objects of different clusters, must be built. Obviously, the similarity measure directly affects the formation of the resulting clusters. This key ingredient of clustering algorithms is called “*distance metric*”. A distance metric  $d$  is a function that takes as arguments two objects  $x$  and  $y$  in an  $n$ -dimensional space  $R^n$ . A lot of distance measures have been considered in literature: some of these are reported and discussed in subsection 1.1.

Once a distance measure is chosen, the construction of a clustering criterion function makes the partition of clusters an optimization problem, which is well defined mathematically, and has rich solutions in the literature. We discuss on this feature in subsection 1.2.

3) **Cluster validation.** As already mentioned at the beginning of the chapter, given a data set, each clustering algorithm can always generate a division of the input objects. Moreover, different approaches usually lead to different cluster and even for the same algorithm, parameter identification or the presentation order of input patterns may affect the final results.

Generally, there are three categories of testing criteria: external indices (that we will call classification-oriented), internal indices (that we will call similarity-oriented), and relative indices. External indices are based on some pre-specified structure, which is the reflection of prior information on the data, and used as a standard to validate the clustering solutions. Internal measures are not dependent on external information (prior knowledge): they observe the clustering structure directly from the original data. We survey some of these indices in subsection 1.4.

4) **Results interpretation.** The ultimate goal of clustering is to provide users with meaningful insights from the original data, so that they can effectively solve the problems encountered. Experts in the relevant fields interpret the data partition. Further analyzes, even experiments, may be required to guarantee the reliability of extracted knowledge.

Cluster analysis is not a one-shot process. In many circumstances, it needs a series of trials and repetitions. Moreover, there are no universal and effective measures to guide the selection of features and clustering schemes. Validation indexes provide some insights on the quality of clustering solutions. But even how to choose the appropriate criterion is still a problem requiring more efforts.

## 1.1 Distance Measures

When a clustering algorithm is designed, it is natural to ask what kind of standards we should use to measure the distance (dissimilarity or similarity) between a pair of objects, an object and a cluster, or a pair of clusters.

Usually, a data object is described by a set of features, represented as a multidimensional vector. These features can be quantitative or qualitative,

continuous or binary, nominal or ordinal which determine the corresponding measure mechanisms.

When a distance function is defined, it must satisfy the following properties:

- a) *Symmetry*: The distance should be symmetric, i.e.:  $d(x, y) = d(y, x)$
- b) *Positivity*: The distance between any two objects should be a real number greater than or equal to zero:  $d(x, y) \geq 0$  for all  $x$  and  $y$ .
- c) *Triangle inequality*: The distance between two objects  $x$  and  $y$  should be shorter than or equal to the sum of the distances from  $x$  to a third object  $z$  and from  $z$  to  $y$ :  $d(x, y) \leq d(x, z) + d(z, y)$  for all  $x, y$  and  $z$ .

In this section, we review different distance measures between two objects defined by two  $n$ -dimensional vectors  $x = (x_1, x_2, \dots, x_n)$  and  $y = (y_1, y_2, \dots, y_n)$ .

### **Euclidean distance**

The most commonly used metric define as:

$$d_E(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1-1)$$

The Euclidean distance is simply the geometric distance. Deriving the Euclidean distance between two data points involves computing the square root of the sum of the squares of the differences between corresponding values.

### **Squared Euclidean distance**

$$d_{E^2}(x, y) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2 = \sum_{i=1}^n (x_i - y_i)^2 \quad (1-2)$$

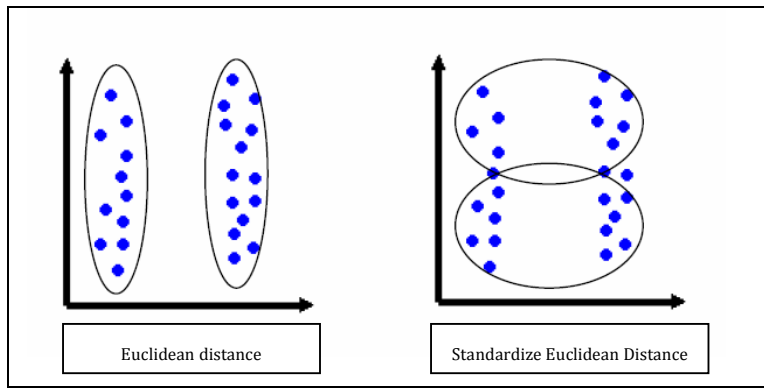
The squared Euclidean distance tends to give more weight to outliers than the Euclidean distance because of the lack of squared root. Data which is clustered using this distance metric might appear sparser and less compact than the Euclidean distance metric. In addition, this metric is more sensitive to miscalculated data than is the Euclidean distance metric.

### **Standardized Euclidean distance**

This distance metric is measured very similar to the Euclidean distance (1-1) except that every dimension is divided by its standard deviation:

$$\begin{aligned} d_{SE}(x, y) &= \sqrt{\frac{1}{s_1^2}(x_1 - y_1)^2 + \frac{1}{s_2^2}(x_2 - y_2)^2 + \dots + \frac{1}{s_n^2}(x_n - y_n)^2} \\ &= \sqrt{\sum_{i=1}^n \frac{1}{s_i^2}(x_i - y_i)^2} \end{aligned} \quad (1-3)$$

This measure gives more importance to dimensions with smaller standard deviation (because of the division by the standard deviation).



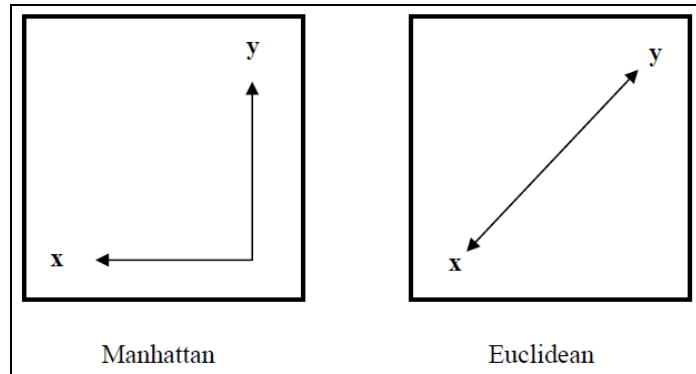
**Fig. 1-2:** An example of better clustering done when using the Standardized Euclidean distance in comparison with the Euclidean distance. The better results are due to equalization of the variances on each axis.

**Manhattan distance**

$$d_M(x, y) = |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n| = \sum_{i=1}^n |x_i - y_i| \quad (1-4)$$

Where  $|x_i - y_i|$  represents the absolute value of the difference between  $x_i$  and  $y_i$ .

The Manhattan distance represents the distance measured along directions that are parallel to the  $x$  and  $y$  axes.



**Fig. 1-3:** The Manhattan vs. Euclidean distance

In Fig. 1-3 it is evident that the Manhattan distance is greater than the Euclidean because of the Pythagorean Theorem.

Data which is clustered using Manhattan distance metric might appear slightly sparser and less compact than the Euclidean distance metric. In addition, this metric is less robust regarding miscalculated data than is the Euclidean distance metric.

**Chebychev distance**

The Chebychev distance will simply pick the maximum absolute difference in values for any objects. This implies that any changes in lower values will be discarded. This kind of metric is very sensitive to outlying values.

$$d_{\max}(x, y) = \max_i |x_i - y_i| \quad (1-5)$$

**Angle between vectors**

$$d_{\alpha}(x, y) = \cos(\theta) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \cdot \sqrt{\sum_{i=1}^n y_i^2}} \quad (1-6)$$

This metric takes into account only the angle and discards the magnitude. Note that if a point is shifted by scaling all its coordinates by the same factors (i.e. noise), the angle distance will not change. This distance is not sensitive to noise if the noise adds some constant value to all dimensions (assuming different values in different dimensions).

**Correlation distance**

$$d_R(x, y) = 1 - p(x, y) \quad (1-7)$$

Where  $p_{xy}$  is the Pearson correlation coefficient of the vectors  $x$  and  $y$ :

$$p(x, y) = \frac{s_{xy}}{\sqrt{s_x} \sqrt{s_y}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1-8)$$

Since the Pearson correlation coefficient  $p(x,y)$  takes values between -1 and 1, the distance  $1 - p(x,y)$  will vary between 0 and 2.

This metric is commonly used in the bioinformatics domain, where the Pearson correlation is used to find whether two differentially expressed genes vary in the same way. For example, the correlation between two genes  $x$  and  $y$ , represented by two vectors containing their expression levels, will be high if the corresponding expression levels increase or decrease at the same time, otherwise the correlation will be low (see Fig. 1-4). In particular, in this figure the black profile and the red profile have almost perfect Pearson correlation despite the differences in basal expression level and scale.

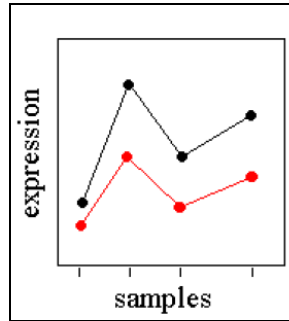


Fig. 1-4: Expression levels in different sample of two genes.

### *Mahalanobis distance*

$$d_m(x, y) = \sqrt{(x - y)^T + S^{-1}(x - y)} \quad (1-9)$$

where  $S$  is any  $n \times n$  positive definite matrix and  $(x-y)^T$  is the transposition of  $(x-y)$ . The role of the matrix  $S$  is to distort the space as desired. It is very similar to what is done with the Standardized Euclidean distance except that the variance may be measured not only along the axes but in any suitable direction. If the matrix  $S$  is taken to be the identity matrix then the Mahalanobis distance reduces to the classical Euclidean distance.

## **1.2 Clustering as an Optimization Problem**

Once a distance measure is chosen, the construction of a clustering criterion function makes the partition of clusters an optimization problem, which is well defined mathematically, and has rich solutions in the literature.

In this way clustering methods correspond to the optimization of some objective global functions and consequently can be treated as an optimization problem.

Given the data objects  $x_i$  belonging to the dataset  $X$  and a set of clusters  $C_j$  with  $j = 1 : J$ , the clustering problem consists in assigning each object  $x_i$  to a cluster  $C_j$  such that the intra-cluster distance is minimized and the inter-cluster distance is maximized.

If we define a matrix  $Z$  of dimension  $J \times J$ , as:

$$z_{ij} = \begin{cases} 1 & \text{if } x_i \in C_j \\ 0 & \text{otherwise} \end{cases} \quad (1-10)$$

The problem can be formulated, in general terms, as:



$$\begin{aligned}
& \min \sum_{j=1}^J \left[ \sum_{i,k=1}^{I \times I} \left[ \text{dist}(x_i, x_k) z_{ij} z_{kj} - \text{dist}(x_i, x_k) z_{ij} (1 - z_{kj}) \right] \right] \\
& \text{s.t.} \tag{1-11} \\
& \sum_{j=1}^J [z_{ij} = 1] \quad \forall i \\
& z_{ij} \in \{0,1\}
\end{aligned}$$

This is a quadratic assignment problem known to be NP-Hard (Gonzalez, 1985) and several heuristics has been proposed to solve it as presented in (Hand et al., 2001).

So, given an objective function such as “minimize SSE (sum of squared error)” clustering can be treated as an optimization problem. One way to solve this problem—to find a global optimum—is to enumerate all possible ways of dividing the points into clusters and then choose the set of clusters that best satisfies the objective function. Of course, this exhaustive strategy is computationally infeasible and as a result, a more practical approach is needed, even if such an approach finds solutions that are not guaranteed to be optimal. One technique, which is known as gradient descent, is based on picking an initial solution and then repeating the following two steps: compute the change to the solution that best optimizes the objective function and then update the solution. The optimization problem presented in this section can be resolved using different clustering algorithms that will be presented in the next session.

### 1.3 Clustering algorithms

Different starting points and criteria usually lead to different taxonomies of clustering algorithms. A rough but widely agreed frame is to classify clustering techniques as partitional clustering and hierarchical clustering, based on the proprieties of cluster generated. In this section, we use the following two simple, but important techniques to introduce many of the concepts involved in cluster analysis. Partitional clustering directly divides data objects into some prespecified number of clusters, while hierarchical clustering groups data objects with a sequence of partitions, either from singleton clusters to a cluster including all individuals or vice versa.

#### 1.3.1 Partitional K-Means Clustering Algorithm

K-means algorithm is a prototype-based, partitional clustering technique that attempts to find a user-specified number of clusters ( $K$ ), which are represented by their centroids.

In principal, the optimal partition, based on some specific criterion, can be found by enumerating all possibilities. But this brute force method is infeasible in practice, due to the expensive computation. Therefore, heuristic algorithms have been developed in order to seek approximate solutions. K-means clustering technique is simple, and aims at assigning a set of object into  $K$  clusters with no hierarchical structure.

We begin with a description of the basic algorithm. We first choose  $K$  initial centroids, where  $K$  is a user specified parameter, namely, the number of clusters desired. Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a *cluster*. The centroid of each cluster is then updated based on the points assigned to the cluster. The assignment and update steps are repeated until no point changes clusters, or equivalently, until the centroids remain the same. K-means is formally described by Algorithm 1-1.

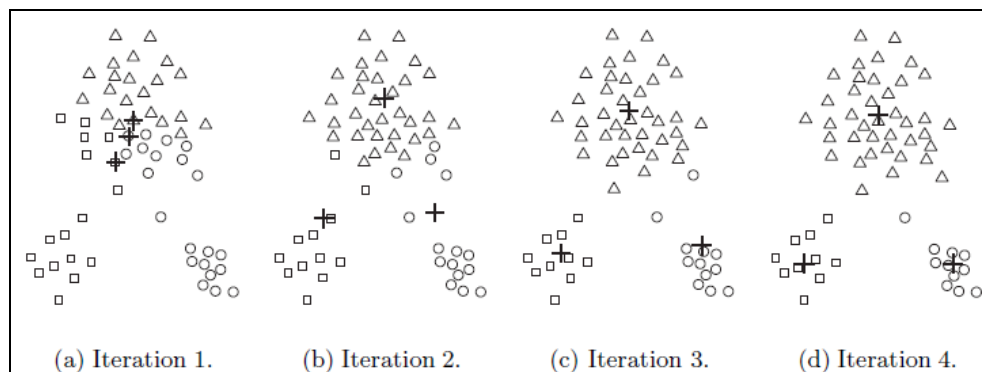
---

**Algorithm 1-1:** Basic k-Means clustering algorithm.

---

- 1: Select  $k$  points as initial centroids.
  - 2: **repeat**
  - 3:   form  $k$  clusters by assigning each point to its closest centroid.
  - 4:   Recomputed the centroid of each cluster.
  - 5: **Until** centroids do not change
- 

The operation of K-means is illustrated in Fig. 1-5, which shows how, starting from three centroids, the final clusters are found in four assignment-update steps.



**Fig. 1-5:** Example of using the K-means algorithm to find three clusters in sample data

In the first step points are assigned to the initial centroids, which are all in the larger group of points. For this example, we use the mean as the centroid. After points are assigned to a centroid, the centroid is then updated. Again, the figure for each step shows the centroid at the beginning of the step and the assignment of points to those centroids. In the second step, points are assigned to the updated centroids, and the centroids are updated again. In steps 2, 3, and 4, which are shown in Fig. 1-5 (b), (c), and (d), respectively, two of the centroids move to the two small groups of points at the bottom of the figures. When the K-means algorithm terminates in Fig. 1-5 (d), because no more changes occur, the centroids have identified the natural groupings of points.

The k-means algorithm is very simple and can be easily implemented in solving many practical problems.

The space requirements are modest because only the data points and centroids are stored. Specifically, the storage required is  $O((m + K)n)$ , where  $m$  is the number of points and  $n$  is the number of attributes.

The time requirements for K-means are also modest basically linear in the number of data points. In particular, the time required is  $O(I * K * m * n)$ , where  $I$  is

the number of iterations required for convergence. As mentioned,  $I$  is often small and can usually be safely bounded, as most changes typically occur in the first iterations. Therefore, K-means is linear in  $m$ , the number of points and is efficient as well as simple provided that  $K$ , the number of clusters, is significantly less than  $m$ . Parallel techniques for k-means are developed that can largely accelerate the algorithm.

The drawbacks of k-means are also well studied, and as a result, many variants of k-means have appeared in order to overcome these obstacles.

Therefore, this algorithm has a long history, but is still the subject of current research. The original K-means algorithm was proposed by (MacQueen et al., 1967). The ISODATA algorithm by (Ball and Hall, 1967) was a premature version of K-means that employed various pre- and post-processing techniques to improve on the basic algorithm. The K-means algorithm and many of its variations are described in detail in the books by (Anderberg, 1983) and (Jain and Dubes, 1988).

The bisecting K-means algorithm (we will outline a version called “Induced bisecting k-means” (Archetti et al., 2006) in the next subsection) was described in (Steinbach et al., 2000), and an implementation of this and other clustering approaches is freely available for academic use in the CLUTO (CLUstering TOolkit) package created by (Karypis et al., 2003).

(Boley et al., 1998) has created a divisive partitioning clustering algorithm (PDDP) based on finding the first principal direction (component) of the data, and (Savaresi and Boley, 2004) has explored its relationship to bisecting K-means. Recent variations of K-means are a new incremental version of K-means (Dhillon et al., 2002), X-means (Pelleg and Moore, 2000), and K-harmonic means (Zhang et al., 1999). While some of the previously mentioned approaches address the initialization problem of K-means in some manner, other approaches to improving K-means initialization can also be found in the work of (Bradley and Fayyad, 1998). (Dhillon and Modha, 2001) present a generalization of K-means, called spherical K-means, that works with commonly used similarity functions. A general framework for K-means clustering that uses dissimilarity functions based on Bregman divergences was constructed by (Banerjee et al., 2004).

### 1.3.1.1 Induced Bisecting K-means

The bisecting K-means algorithm, proposed in (Savaresi et al., 2001; Steinbach et al., 2000), is a straightforward extension of the basic K-means algorithm that is based on a simple idea: to obtain  $K$  clusters, split the set of all points into two clusters, select one of these clusters to split, and so on, until  $K$  clusters have been produced. In particular, bisecting k-Means has a linear complexity and is relatively efficient and scalable.

It starts with a single cluster of all input points and works as reported in Algorithm 1-2:

---

**Algorithm 1-2:** Bisecting k-Means algorithm.

---

- 1: Initialize the list of clusters to contain the cluster consisting of all points.
- 2: **repeat**
- 3:   remove a cluster  $S$  from the list of clusters
- 4:   **for**  $i=1$  to  $ITER$  **do**
- 5:     Select two random seeds which are the initial centroids
- 6:     Bisect the selected cluster  $S$  using basic K-Means
- 7:   **end for**
- 8:   Select the two clusters from the bisection with the highest Intra Cluster Similarity (ICS)

$$ICS_{S_k} = \frac{1}{|S_k|^2} \sum_{\substack{d \in S_k \\ d' \in S_k}} \cos(d, d') \quad (1-12)$$

- 9:   Add these two clusters to the list of clusters
  - 10: **Until** the list of clusters contains  $K$  clusters
- 

The major disadvantage of this algorithm is that it requires the a priori specification of  $K$  and  $ITER$  parameters. An incorrect estimation of  $K$  and  $ITER$  may lead to poor clustering accuracy. Moreover, the algorithm is sensitive to the noise which may affect the computation of cluster centroids. For any given cluster let  $N$  be the number of objects belonging to that cluster and  $R$  the set of their indices. In fact, the  $j^{\text{th}}$  element of a cluster centroid is computed as:

$$q_j = \frac{1}{N} \sum_{r \in R} m_{rj} \quad (1-13)$$

where  $N$  represents the number of objects belonging to the cluster.

The centroid  $c$  may contain also the contribution of noisy terms contained in the objects which the pre-processing phase and feature selection phase have not been able to remove. To overcome these two problems an extended version of the Standard Bisecting k-Means, named *Induced Bisecting k-Means*, has been proposed in (Archetti et al., 2006). Its main steps are described as follows:

1. Set the Intra Cluster Similarity (ICS) threshold parameter  $\tau$
2. Build a distance matrix  $A$ , of dimension  $|Q| \times |Q|$ , whose elements are given by the Euclidean distance between objects

$$a_{ij} = \sqrt{\sum_{k=1}^{|Q|} (m_{ik} - m_{jk})^2} \quad (1-14)$$

where  $i, j \in Q$ .

3. Select, as centroids, the two objects  $i$  and  $j$  s.t.

$$a_{ij} = \max_{l, m = \{1, \dots, |Q|\}} A_{lm} \quad (1-15)$$

The splitting is also different from the Standard Bisecting k-Means and is performed according to the following 3 steps:

4. Find 2 sub-clusters  $S_1$  and  $S_2$  using the basic k-Means algorithm.
5. Check the ICS of  $S_1$  and  $S_2$  as:

- a. If the ICS value of a cluster is smaller than  $\tau$  then reapply the divisive process to this set, starting from step 2.
  - b. If the ICS value of a cluster is over a given threshold, then stop.
6. The entire process will finish when there are no sub-clusters to divide.

The main differences of this algorithm with respect to the Standard Bisecting k-Means consist in:

- how the initial centroids are chosen: as centroids of the two child clusters we select the objects of the parent cluster having the greatest distance between them.
- the cluster splitting rule: a cluster is split in two if its Intra Cluster Similarity is smaller than a threshold parameter  $\tau$ . Therefore, the “optimal” number of cluster  $K$  is controlled by the parameter  $\tau$ . The main advantages being that no input parameters  $K$  and ITER must be specified by the user.

### 1.3.2 Hierarchical Clustering Algorithm

Hierarchical clustering (HC) techniques are a second important category of clustering methods. These algorithms organize data into a hierarchical structure according to the similarity matrix. As K-means, these approaches are relatively old, but they still enjoy widespread use.

Much of the initial activity was in the area of taxonomy and is covered in books by (Sneath and Sokal, 1971; Jardine and Sibson et al 1988). Agglomerative hierarchical clustering is the focus of most work in the area of hierarchical clustering, but divisive approaches have also received some attention. For example, (Zahn et al., 1971) describes a divisive hierarchical technique that uses the minimum spanning tree of a graph.

There are two basic approaches for generating a hierarchical clustering:

- *Agglomerative*: Start with the points as individual clusters and, at each step, merge the closest pair of clusters. This requires defining a notion of cluster proximity.
- *Divisive*: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide which cluster to split at each step and how to do the splitting.

While both divisive and agglomerative approaches typically take the view that merging (splitting) decisions are final, there has been some work, which we shall not consider here, by (Fisher et al., 1996) and (Karypis et al., 1999) to overcome these limitations.

Agglomerative hierarchical clustering techniques are by far the most common, and, in this section, we will focus exclusively on these methods.

The results of hierarchical clustering are usually depicted by a binary tree or dendrogram which displays both the relationships between cluster and sub-cluster and the order in which the clusters were merged (agglomerative view) or split (divisive view). The root node of the dendrogram represents the whole data set and each leaf node is regarded as a data object. The intermediate nodes, thus, describe the extent that the objects are proximal to each other; and the height of the dendrogram usually expresses the distance between each pair of objects or

clusters, or an object and a cluster. The ultimate clustering results can be obtained by cutting the dendrogram at different levels.

This kind of representation provides very informative descriptions and visualization for the potential data clustering structures, especially when real hierarchical relations exist in the data, like the data from evolutionary research on different species of organisms.

Usually, for sets of two-dimensional points the graphical representation of hierarchical clustering is made using a nested clustering diagram, like this in Fig. 1-6 that shows an example of the two types of figures for a set of four two-dimensional points ( $p_1, p_2, p_3, p_4$ ).

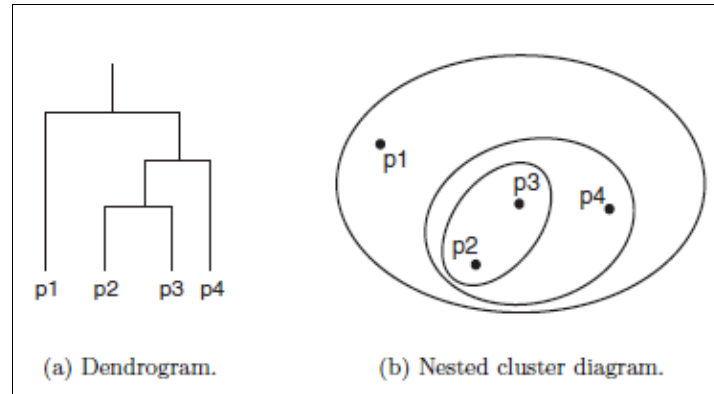


Fig. 1-6: A hierarchical clustering of four points shown as a dendrogram and as nested clusters

### 1.3.2.1 Basic Agglomerative Hierarchical Clustering Algorithm

The key process at the basis of many agglomerative hierarchical clustering techniques can be summarized with few simple steps: starting with individual points as clusters, successively merge the two closest clusters until only one cluster remains. A more formally description of this algorithm is expressed in Algorithm 1-3.

---

**Algorithm 1-3:** Basic agglomerative hierarchical clustering algorithm.

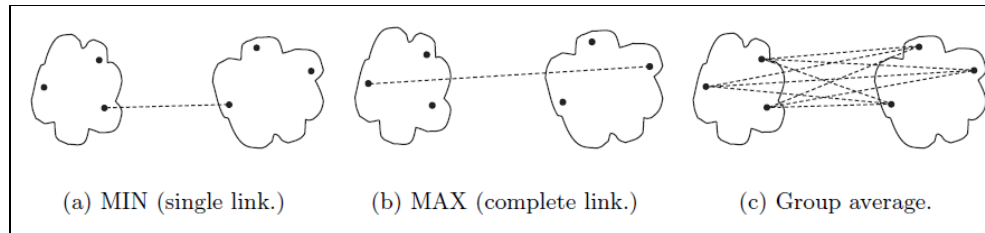
---

- 1: Compute the proximity matrix
  - 2: **repeat**
  - 3:   merge the closest two clusters
  - 4:   Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
  - 5: **Until** only one cluster remains
- 

The computation of the proximity between two clusters, typically defined with a particular type of cluster in mind, is the key operation of Algorithm 1-3. For example, many agglomerative hierarchical clustering techniques, such as *MIN*, *MAX*, and *Group Average*, come from a graph-based view of clusters. *MIN* defines cluster proximity as the proximity between the closest two points that are in different clusters, or using graph terms, the shortest edge between two nodes in different subsets of nodes. This yields contiguity-based clusters.

Alternatively, *MAX* takes the proximity between the farthest two points in different clusters to be the cluster proximity, or using graph terms, the longest

edge between two nodes in different subsets of nodes. (If our proximities are similarities, where higher values indicate closer points, is usually preferred to use *single link* and *complete link*). Another graph-based approach, the *group average* technique, defines cluster proximity to be the average pair wise proximities (average length of edges) of all pairs of points from different clusters. Fig. 1-7 illustrates these three approaches.



**Fig. 1-7:** Graph-based definitions of cluster proximity

The basic agglomerative hierarchical clustering algorithm just presented uses a proximity matrix. This requires the storage of  $\frac{1}{2}m^2$  proximities (assuming the proximity matrix is symmetric) where  $m$  is the number of data points.

The space needed to keep track of the clusters is proportional to the number of clusters, which is  $m-1$ , excluding singleton clusters. Hence, the total space complexity is  $O(m^2)$ .

The analysis of the basic agglomerative hierarchical clustering algorithm is also straightforward with respect to computational complexity.  $O(m^2)$  time is required to compute the proximity matrix. After that step, there are  $m-1$  iterations involving steps 3 and 4 because there are  $m$  clusters at the start and two clusters are merged during each iteration. If performed as a linear search of the proximity matrix, then for the  $i$ th iteration, step 3 requires  $O((m-i+1)^2)$  time, which is proportional to the current number of clusters squared. Step 4 only requires  $O(m-i+1)$  time to update the proximity matrix after the merger of two clusters. Without modification, this would yield a time complexity of  $O(m^3)$ . If the distances from each cluster to all other clusters are stored as a sorted list (or heap), it is possible to reduce the cost of finding the two closest clusters to  $O(m-i+1)$ . However, because of the additional complexity of keeping data in a sorted list or heap, the overall time required for a hierarchical clustering based on Algorithm 1-3 is  $O(m^2 \log m)$ .

The space and time complexity of hierarchical clustering severely limits the size of data sets that can be processed.

## 1.4 Different Measures of Cluster Validation

In supervised classification, the evaluation of the resulting classification model is an integral part of the process of developing a classification model, and there are well-accepted evaluation measures and procedures, e.g., accuracy and cross-validation, respectively. However, because of its very nature, cluster evaluation is not a well-developed or commonly used part of cluster analysis.

Nonetheless, cluster evaluation, or cluster validation as it is more traditionally called, is important; many times, cluster analysis is conducted as a part of an exploratory data analysis.

When we have external information about data, it is typically in the form of externally derived class labels for the data objects. In such cases, the usual procedure is to measure the degree of correspondence between the cluster labels and the class labels. Motivations for such an analysis are the comparison of clustering techniques with the “ground truth” or the evaluation of the extent to which a manual classification process can be automatically produced by cluster analysis.

In next subsections we present different cluster validity measures for both partitional clustering and hierarchical clustering.

### 1.4.1 Cluster Validity Measures for Partitional Clustering

We consider two different kinds of approaches:

- *External validity measures*: a set of techniques that use measures from classification, such as entropy, purity, and the F-measure. These measures evaluate the extent to which a cluster contains objects of a single class.
- *Internal validity measures*: a group of methods related to the similarity measures. These approaches measure the extent to which two objects that are in the same class are in the same cluster and vice versa.

For convenience, we will refer to the external validity measures as *classification-oriented* and to the internal validity measures as *similarity-oriented*.

#### 1.4.1.1 Classification-Oriented Measures of Cluster Validity

There are a number of measures, like entropy, purity, precision, recall and the F-measure, that are commonly used to evaluate the performance of a classification model. In the case of classification, we measure the degree to which predicted class labels correspond to actual class labels, but for the measures just mentioned, nothing fundamental is changed by using cluster labels instead of predicted class labels. Next, we quickly review the definitions of these measures.

##### **Entropy**

The degree to which each cluster consists of objects of a single class. For each cluster, the class distribution of the data is calculated first, i.e., for class  $j$  we compute  $p_{ij}$ , the probability that a member of cluster  $i$  belongs to class  $j$  as:

$$p_{ij} = \frac{m_{ij}}{m_i} \quad (1-16)$$

where  $m_i$  is the number of objects in cluster  $i$  and  $m_{ij}$  is the number of objects of class  $j$  in cluster  $i$ . Using this class distribution, the entropy of each cluster  $i$  is calculated using the standard formula

$$e_i = -\sum_{j=1}^L p_{ij} \log_2 p_{ij} \quad (1-17)$$

where  $L$  is the number of classes.

The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e.:



$$e = \sum_{i=1}^K \frac{m_i}{m} e_i \quad (1-18)$$

where  $K$  is the number of clusters and  $m$  is the total number of data points.

### **Purity**

Another measure of the extent to which a cluster contains objects of a single class. Using the previous terminology, the purity of cluster  $i$  is:

$$p_i = \max_j p_{ij} \quad (1-19)$$

the overall purity of a clustering is:

$$purity = \sum_{i=1}^K \frac{m_i}{m} p_i \quad (1-20)$$

where  $m_i$  is the number of data points of cluster  $i$ .

### **Precision**

The fraction of a cluster that consists of objects of a specified class. The precision of cluster  $i$  with respect to class  $j$  is:

$$precision(i, j) = p_{ij} \quad (1-21)$$

### **Recall**

The extent to which a cluster contains all objects of a specified class. The recall of cluster  $i$  with respect to class  $j$  is:

$$recall(i, j) = \frac{m_{ij}}{m_j} \quad (1-22)$$

where  $m_j$  is the number of objects in class  $j$ .

### **F-measure**

A combination of both precision and recall that measures the extent to which a cluster contains only objects of a particular class and all objects of that class. The F-measure of cluster  $i$  with respect to class  $j$  is

$$F(i, j) = \frac{(2 \times precision(i, j) \times recall(i, j))}{(precision(i, j) + recall(i, j))} \quad (1-23)$$

## **1.4.1.2 Similarity-Oriented Measures of Cluster Validity**

### ***F* statistic measures**

We can view this approach to cluster validity as involving the comparison of two matrices:

- (1) the ideal cluster similarity matrix, which has a 1 in the  $ij^{th}$  entry if two objects,  $i$  and  $j$ , are in the same cluster and 0, otherwise
- (2) an ideal class similarity matrix defined with respect to class labels, which has a 1 in the  $ij^{th}$  entry if two objects,  $i$  and  $j$ , belong to the same class, and a 0 otherwise. As before, we can take the correlation of these two matrices as the measure of cluster validity.

*Example: Correlation between Cluster and Class Matrices:*

To demonstrate this idea more concretely, we give an example involving five data points,  $p_1, p_2, p_3, p_4, p_5$ , two clusters,  $C_1 = \{p_1, p_2, p_3\}$  and  $C_2 = \{p_4, p_5\}$ , and two classes,  $L_1 = \{p_1, p_2\}$  and  $L_2 = \{p_3, p_4, p_5\}$ .

The ideal cluster and class similarity matrices are given in Tab. 1-1 and Tab. 1-2. The correlation between the entries of these two matrices is 0.359.

Point	p <sub>1</sub>	p <sub>2</sub>	p <sub>3</sub>	p <sub>4</sub>	p <sub>5</sub>
p <sub>1</sub>	1	1	1	0	0
p <sub>2</sub>	1	1	1	0	0
p <sub>3</sub>	1	1	1	0	0
p <sub>4</sub>	0	0	0	1	1
p <sub>5</sub>	0	0	0	1	1

**Tab. 1-1:** Ideal cluster similarity matrix

Point	p <sub>1</sub>	p <sub>2</sub>	p <sub>3</sub>	p <sub>4</sub>	p <sub>5</sub>
p <sub>1</sub>	1	1	0	0	0
p <sub>2</sub>	1	1	0	0	0
p <sub>3</sub>	0	0	1	1	1
p <sub>4</sub>	0	0	1	1	1
p <sub>5</sub>	0	0	1	1	1

**Tab. 1-2:** Ideal classification similarity matrix

More generally, we can use any of the measures for binary similarity (For example, we can convert these two matrices into binary vectors by appending the rows). Specifically, we need to compute the following four quantities for all pairs of distinct objects (There are  $m(m - 1)/2$  such pairs, if  $m$  is the number of objects):

$f_{00}$  = number of pairs of objects having a different class and a different cluster

$f_{01}$  = number of pairs of objects having a different class and the same cluster

$f_{10}$  = number of pairs of objects having the same class and a different cluster

$f_{11}$  = number of pairs of objects having the same class and the same cluster

In particular, the simple matching coefficient, which is known as the *Rand statistic* in this context, and the *Jaccard coefficient* are two of the most frequently used cluster validity measures.

$$\text{Rand statistic} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}} \quad (1-24)$$

$$\text{Jaccard coefficient} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} \quad (1-25)$$

*Example: Rand and Jaccard Measures:*

Based on these formulas, we can readily compute the Rand statistic and Jaccard coefficient for the example based on Tab. 1-1 and Tab. 1-2. Noting that  $f_{00} = 4$ ,  $f_{01} = 2$ ,  $f_{10} = 2$ , and  $f_{11} = 2$ , the Rand statistic =  $(2 + 4)/10 = 0.6$  and the Jaccard coefficient =  $2/(2+2+2) = 0.33$ . We also note that the four quantities,  $f_{00}$ ,  $f_{01}$ ,  $f_{10}$  and  $f_{11}$ , define a *contingency table* as shown in Tab. 1-3.

	Same Cluster	Different Cluster
Same Class	$f_{11}$	$f_{10}$
Different Class	$f_{01}$	$f_{00}$

**Tab. 1-3:** Two-way contingency table for determining whether pairs of objects are in the same class and cluster.

### DVI measure

The DVI index, proposed by (Shen et al., 2005) is based on an intra/inter ratio validity index that also includes scaling of the intra- and the inter-cluster distances.

$$DVI_k = \min \left\{ \frac{\text{intra}(k)}{\max_{i=2,\dots,K} \{\text{intra}(i)\}} + \frac{\text{inter}(k)}{\max_{i=2,\dots,K} \{\text{inter}(i)\}} \right\} \quad (1-26)$$

where:

$$\frac{\text{intra}(k)}{\max_{i=2,\dots,K} \{\text{intra}(i)\}} = \text{IntraRatio}(k) \quad \frac{\text{inter}(k)}{\max_{i=2,\dots,K} \{\text{inter}(i)\}} = \text{InterRatio}(k) \quad (1-27)$$

and

$$\text{intra}(k) = \frac{1}{N} \sum_{i=1}^k \sum_{x \in C_i} \|x - q_i\|^2 \quad (1-28)$$

$$\text{inter}(k) = \frac{\text{Max}_{i,j} (\|q_i - q_j\|^2)}{\text{Min}_{i \neq j} (\|q_i - q_j\|^2)} \sum_{i=1}^k \left( \frac{1}{\sum_{j=1}^k (\|q_i - q_j\|^2)} \right) \quad (1-29)$$

where  $k$  is the number of clusters,  $N$  is the number of data points and  $\text{intra}$  is the average Euclidean distance between data points and the prototype  $q$  of the cluster  $C_i$  each sample is assigned to.

In the above equations, *IntraRatio* is used to represent the overall compactness of clusters which is scaled from Intra term; *InterRatio* is used to represent the overall separateness of clusters which is scaled from Inter term. The normalized ratios are used for the purpose of comparison.

The *Intra* term is defined as the average sum square of the distance from the data points to the cluster centroids. The value of the Intra term generally decreases with cluster number since the clusters become more and more compact. Thus, the normalized intra-ratio is the same and its value ranges between 0 and 1.

The *Inter* term is composed of two parts:  $\frac{\text{Max}_{i,j} (\|q_i - q_j\|^2)}{\text{Min}_{i \neq j} (\|q_i - q_j\|^2)}$  and  $\sum_{i=1}^k \left( \frac{1}{\sum_{j=1}^k (\|q_i - q_j\|^2)} \right)$ , both

of them are influenced by the geometry of the cluster centroids.

The value of *Inter* tends to increase with the number of clusters,  $k$ . The Inter term is more sensitive to the distance between clusters than the Intra term.

So, the *DVI Index* should be more significant when the clusters is merged or split. Since the Inter term is more sensitive to the distance between clusters than the Intra term, one modulating parameter  $\gamma$  could be used to balance the importance between the *IntraRatio* and *InterRatio* terms. Usually, this parameter is set  $\gamma = 1$  if there is no noise in the raw data. If exists some noise in the data, the effect of such noise could be decreased by adjusting the parameter  $\gamma$  less than 1; and this parameter  $\gamma$  could also be adjusted to be greater than 1 in some special cases where the within-cluster compactness is more important than the between-cluster separateness.

In general, this index is used also to find the cluster number  $k$ . In fact the relationship between DVI and the cluster number  $k$  is simplified as *DVI Index* =  $f(k)$  if no other factors are included into this function. In other words, the optimal cluster number is obtained when the DVI Index value reaches its minimum where the  $k$  value is considered a good indication for the “true” number of clusters in the data set.

### **Categories Utility measure**

The *Categories Utility* (CU) measure (Gluck et al., 1985), used only for categorical data, defines the probability of matching a categorical feature value given a cluster versus the probability of the categorical feature value given the entire data set

$$CU_m = \frac{1}{K} \sum_{k=1}^K P(C_k) \left[ \sum_i \sum_j P(x_i = v_{ij} | C_k)^2 - \sum_i \sum_j P(x_i = v_{ij})^2 \right] \quad (1-30)$$

Where:

- $P(A_i = V_{ij})$  is the unconditional probability of feature  $x_i$  taking on the value  $v_{ij}$
- $P(A_i = V_{ij} | C_k)$  is the conditional probability of  $A_i = V_{ij}$  given cluster  $C_k$
- $k$  is the cluster number from 1 to  $K$ .

### **DVI\_CU measure**

For mixed data types, (Bushel et al., 2007) proposed an index that combines both DVI index for numerical data and CU index for categorical data.

The *DVI* modified with *CU*.

$$DVI\_CU = DVI + \frac{1}{CU} \quad (1-31)$$

This index is minimized over all  $k$  sets for each run of the modk-prototypes clustering algorithm.

## **1.4.2 Cluster Validity Measures for Hierarchical Clustering**

So far in this section, we have discussed supervised measures of cluster validity only for partitional clustering. Supervised evaluation of a hierarchical clustering is more difficult for a variety of reasons, including the fact that a pre-existing hierarchical structure often does not exist. Here, we will give an example of an approach for evaluating a hierarchical clustering in terms of a (flat) set of class labels, which are more likely to be available than a pre-existing hierarchical structure.

The key idea of this approach is to evaluate whether a hierarchical clustering contains, for each class, at least one cluster that is relatively pure and includes most of the objects of that class. To evaluate a hierarchical clustering with respect to this goal, we compute, for each class, the F-measure for each cluster in the cluster hierarchy. For each class, we take the maximum F-measure attained for any cluster. Finally, we calculate an overall F-measure for the hierarchical clustering by computing the weighted average of all per-class F-measures, where the weights are based on the class sizes. More formally, this hierarchical F-measure is defined as follows:

$$F = \sum_j \frac{m_j}{m} \max_i F(i, j) \quad (1-32)$$

where the maximum is taken over all clusters  $i$  at all levels,  $m_j$  is the number of objects in class  $j$ , and  $m$  is the total number of objects.

## 2. Specific Challenges Related to Biomedical Data

A suitable clustering algorithm depends on the application and data type (Han and Kamber, 2006).

Clustering with its most popular algorithms, partitioned k-means and hierarchical (presented in section 1.3.1 and 1.3.2) is universally applied in life science domain. Numerous improvements of these two traditional methods have been introduced, as well as totally different approaches.

Like said in (Andreopoulos et al., 2009), clustering in life science involves principally two groups of users, both of which need to recognize what algorithmic features a biological application requires. One user group includes biologists with experience on the underlying biological problem, who apply existing clustering algorithms to solve the problem. The challenge is to choose a suitable algorithm, since each algorithm will produce different results. For instance, in clustering gene expression data a biologist wishes to mix numerical expression levels with discrete Gene Ontology (GO) categorization.

Another user group includes computer scientists who develop novel bioinformatics algorithms. This group assumes current algorithms are insufficient for the underlying biological problem, and that progress requires improved methods. There is significant overlap between these two user groups, since applications often stimulate algorithmic development.

As a result, requirements and desirable features of biomedical clustering applications are: minimum user-specified input parameters, robustness to noise and outliers, mixed data types and integration of background knowledge (such as Gene Ontology annotations).

In this section we explain and verify three particular challenges that we try to overcome with different proposed methods. These challenges are: “feature selection” (subsection 2.1) to reduce high dimensional (thousands or millions of records with tens or hundreds of attributes) input space and remove noise from data; “mixed data types” (subsection 2.2) to handle both numeric and categorical values (subsection 2.3); finally, “knowledge integration” (subsection 2.3) in order to improve the semantic value of clustering.

### 2.1 Feature Selection

Feature selection (FS) is the process of identifying and removing as much irrelevant and redundant information as possible. The reduction of the dimensionality of the data allows learning algorithms to operate faster and more effectively.

Feature selection is an important tool in many life sciences studies. Given the large complexity of biological data, e.g. the number of genes in a microarray experiment, one naturally looks for a small subset of features (e.g. small number of genes) that may explain the properties of the data that are being investigated.

During the last decade, the motivation for applying feature selection (FS) has become a real prerequisite for model building (Saeys et al., 2007). In particular, the high dimensional nature of many modelling tasks in bioinformatics, going from sequence analysis over microarray analysis to spectral analyses and

literature mining has given rise to a wealth of feature selection techniques being presented in the field. In contrast to other dimensionality reduction techniques like those based on projection (e.g. principal component analysis) or compression (e.g. using information theory), feature selection techniques do not alter the original representation of the variables, but only select a subset of them. Thus, they conserve the original semantics of the variables, hence, offering the advantage of interpretability by a domain expert.

As many pattern recognition techniques were originally not designed to cope with large amounts of irrelevant features, combining them with FS techniques has become a necessity in many applications (Guyon and Elisseeff, 2003; Liu and Motoda, 1998).

Feature selection techniques are usually applied to:

- improve model performance, i.e. prediction performance in the case of supervised classification and better cluster detection in the case of clustering
- provide faster and more cost-effective models, i.e. reducing noise into data
- gain a deeper insight into the underlying processes that generated the data.

However, the advantages of feature selection techniques come at a certain price, as the search for a subset of relevant features introduces an additional layer of complexity in the modelling task, like it's just said in section 1 and in particular it's visible in Fig. 1-1.

Feature selection techniques differ from each other in the way they incorporate the search of finding the optimal subset of relevant features and in the model selection.

Feature selection can be applied to both supervised and unsupervised learning. In the following subsections we focus our attention on both problems: initially on supervised learning (classification), where the class labels are known beforehand, then on the interesting topic of feature selection for unsupervised learning (clustering), a more complex issue that get more attention in several communities (Varshavsky et al., 2006).

### **2.1.1 Supervised Feature Selection**

In the context of supervised machine learning techniques, feature selection techniques can be organized into three categories: *filter methods*, *wrapper methods* and *embedded methods*. The categorization depends on how each technique combine the feature selection search with the construction of the supervised classification model. Fig. 2-1, taken from (Saeys et al., 2007) provides a common taxonomy of feature selection methods. For each feature selection type are highlighted advantages, disadvantages and some examples with the relative references.


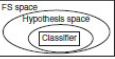
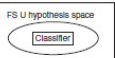
Model search	Advantages	Disadvantages	Examples
<b>Filter</b>  	<b>Univariate</b>  Fast Scalable Independent of the classifier	Ignores feature dependencies Ignores interaction with the classifier	$\chi^2$ Euclidean distance $t$ -test Information gain, Gain ratio (Ben-Bassat, 1982)
	<b>Multivariate</b>  Models feature dependencies Independent of the classifier Better computational complexity than wrapper methods	Slower than univariate techniques Less scalable than univariate techniques Ignores interaction with the classifier	Correlation-based feature selection (CFS) (Hall, 1999) Markov blanket filter (MBF) (Koller and Sahami, 1996) Fast correlation-based feature selection (FCBF) (Yu and Liu, 2004)
<b>Wrapper</b>  	<b>Deterministic</b>  Simple Interacts with the classifier Models feature dependencies Less computationally intensive than randomized methods	Risk of over fitting More prone than randomized algorithms to getting stuck in a local optimum (greedy search) Classifier dependent selection	Sequential forward selection (SFS) (Kittler, 1978) Sequential backward elimination (SBE) (Kittler, 1978) Plus $q$ take-away $r$ (Ferri <i>et al.</i> , 1994) Beam search (Siedlecky and Sklansky, 1988)
	<b>Randomized</b>  Less prone to local optima Interacts with the classifier Models feature dependencies	Computationally intensive Classifier dependent selection Higher risk of overfitting than deterministic algorithms	Simulated annealing Randomized hill climbing (Skalak, 1994) Genetic algorithms (Holland, 1975) Estimation of distribution algorithms (Inza <i>et al.</i> , 2000)
<b>Embedded</b>  	Interacts with the classifier Better computational complexity than wrapper methods Models feature dependencies	Classifier dependent selection	Decision trees Weighted naive Bayes (Duda <i>et al.</i> , 2001) Feature selection using the weight vector of SVM (Guyon <i>et al.</i> , 2002; Weston <i>et al.</i> , 2003)

Fig. 2-1: Taxonomy of feature selection techniques.

*Filter techniques*, looking only at the intrinsic properties of the data, assess the relevance of features. Features' relevance is measured using a "feature relevance score" and, consequently, low-scoring features are removed. Afterwards, the subset of chosen relevant features is presented as input to the classification algorithm. Advantages of these techniques are that they easily scale to very high-dimensional datasets, they are computationally simple and fast, and they are independent of the classification algorithm. As a result, feature selection needs to be performed only once, and then different machine learning methods can be evaluated. A common disadvantage of filter methods is that they ignore the interaction with the classifier (the search in the feature subset space is separated from the search in the hypothesis space). This means that each feature is considered separately, thereby ignoring feature dependencies, which may lead to worse classification performance when compared to other types of feature selection techniques. In order to overcome the problem of ignoring feature dependencies, a number of multivariate filter techniques were introduced, aiming at the incorporation of feature dependencies to some degree.

*Wrapper methods* embed the model hypothesis search within the feature subset search. In this setup, a search procedure in the space of possible feature subsets is defined, and various subsets of features are generated and evaluated. The evaluation of a specific subset of features is obtained by training and testing a



specific classification model, rendering this approach tailored to a specific classification algorithm. To search the space of all feature subsets, a search algorithm is then 'wrapped' around the classification model. However, as the space of feature subsets grows exponentially with the number of features, heuristic search methods are used to guide the search for an optimal subset. Advantages of wrapper approaches include the interaction between feature subset search and model selection, and the ability to take into account feature dependencies. A common drawback of these techniques is that they have a higher risk of over fitting than filter techniques and are very computationally intensive, especially if building the classifier has a high computational cost.

Finally, in *embedded techniques*, the search for an optimal subset of features is built into the classifier construction, and can be seen as a search in the combined space of feature subsets and hypotheses. Just like wrapper approaches, embedded approaches are thus specific to a given learning algorithm. Embedded methods have the advantage that they include the interaction with the classification model, while at the same time being far less computationally intensive than wrapper methods

### 2.1.2 Unsupervised Feature Selection

As just said in the last subsection, many methods have been developed for selecting small informative feature subsets in large noisy data. However, unsupervised methods are limited. Examples are using the variance of data collected for each feature, or the projection of the feature on the first principal component.

Unsupervised feature selection algorithms belong to the field of unsupervised learning. These algorithms are quite different from the major bulk of feature selection studies that are based on supervised methods (e.g., Liu et al., 2002; Guyon and Elisseeff, 2003), and compared to the latter are relatively overlooked. Unsupervised studies, unaided by objective functions, may be more difficult to carry out; nevertheless they convey several important advantages:

- they are unbiased, by neither the experimental expert nor by the data-analyst, can be performed well when no prior knowledge is available,
- they reduce the risk of over fitting (in contrast to supervised feature selection that may be unable to deal with a new class of data).

The downside of the unsupervised approach is that it relies on some mathematical principle and no guarantee is given that this principle is universally valid for all data. A common practice to resolve this quandary is to demonstrate the success of the method on various biological datasets and compare the results obtained by the method with external knowledge.

In this case existing methods can be classified in two principal categories: *wrapper* and *filter*.

*Wrapper methods* contain a well-specified objective function, which should be optimized through the selection. The algorithmic process usually involves several iterations until a target or convergence is achieved.

*Feature filtering* is a process of selecting features without referring back to the data classification or any other target function. Hence we find filtering as a more

suitable process that may be applied in an unsupervised manner. Existing methods of unsupervised feature filtering include ranking of features according to range or variance (e.g., Herrero, 2003, Guyon and Elisseeff, 2003), selection according to highest rank of the first principal component ('Gene shaving' of Hastie et al. 2000, Ding 2003) and other statistical criteria. An example of the latter is (Ben-Dor et al., 2001) where all possible partitions of the data are considered and the corresponding features are labelled. The partitions with statistical significant overabundance are selected. Another example is of (Wolf et al., 2005), who optimize a function based on the spectral properties of the Laplacian of the features.

### 2.1.3 Proposed approach:

#### Genetic Programming for Feature Selection

In this section we focus the attention on a peculiar machine learning method, namely *Genetic Programming (GP)*, which has been shown has good results in the analysis of different life sciences datasets (Archetti et al., 2009a; Archetti et al., 2009b; Vanneschi et al., 2009).

Evolutionary algorithms (defined in artificial intelligence, as a subset of evolutionary computation, a generic population-based metaheuristic optimization algorithm) have also been used for feature selection (Siedlecki and Sklansky, 1988; Casillas et al., 2001; Pal et al., 1998; Sherrah et al., 1996). Usually, in a genetic algorithm based feature selection approach (Siedlecki and Sklansky, 1989), each individual (chromosome) of the population represents a feature subset. For an  $n$ -dimensional feature space, each individual is encoded by an  $n$ -bit binary string  $b_1, \dots, b_n$  where  $b_i=1$  if the  $i$ -th feature is present in the feature subset represented by the individual and  $b_i=0$  otherwise. Therefore, a machine learning algorithm, usually classification, is used to evaluate each individual (or feature subset). Typically each individual is evaluated based on the classification accuracy and the dimension of the feature subset (number of 1s). (Kudo and Sklansky, 2000) in their work affirm that genetic algorithm based feature selection performs better than many conventional feature selection techniques for high-dimensional data. (Siedlecki and Sklansky, 1989) used branch and bound technique for feature selection using genetic algorithms. (Casillas et al., 2001) developed a genetic feature selection scheme for fuzzy rule based classification systems. (Pal et al., 1998) introduced a new particular genetic operator for feature selection: self-crossover.

However, there have been only a few attempts to use genetic programming (Koza, 1992; Banzhaf et al., 1998) for feature selection.

In this section we focus on different applications of Genetic Programming technique to different life science datasets and we will point out strength of the Genetic programming approach as a feature selection technique.

In particular, initially in subsection 2.1.3.1, we proposed a description of the implemented genetic programming framework, subsequently we report all experimental results obtained on three different life science datasets: NCI60 dataset (described in Appendix A1.2), two Oncologic Datasets (briefly described in Appendix A1.4) and finally a molecular dataset (described in Appendix A1.5). For each dataset Genetic Programming has been applied not only for feature

selection scope, but also for prediction (NCI60 and molecular dataset) and classification (two oncologic datasets).

### 2.1.3.1 Genetic Programming Framework

Genetic Programming (GP) (Koza, 1992) is an evolutionary approach which extends Genetic Algorithms (GAs) (Holland, 1975; Goldberg, 1989) to the space of programs. Like any other evolutionary algorithm, GP works by defining a goal in the form of a quality criterion (or fitness) and then using this criterion to evolve a set (also called population) of solution candidates (also called individuals) by mimic the basic principles of Darwin evolution theory. The most common version of GP, and also the one used here, considers individuals as LISP-like tree structures that can be built recursively from a set of function symbols  $F=\{f_1, f_2, \dots, f_n\}$  (used to label internal tree nodes) and a set of terminal symbols  $T=\{t_1, t_2, \dots, t_m\}$  (used to label tree leaves). GP breeds these solutions to solve problems by executing an iterative process involving the probabilistic selection of the fittest solutions and their variation by means of a set of genetic operators, usually crossover and mutation.

#### *Genetic Programming for regression: application to NCI60 and molecular datasets*

Both the application of Genetic Programming to NCI60 dataset and molecular dataset can be regarded as a regression problem.

In particular for NCI60 dataset we look for a relationship between gene expressions and responses to oncology drugs Fluorouracil, Fludarabine, Floxuridine and Cytarabine, i.e. we aim at identifying, from genomic measurements of biopsies, the likelihood to develop drug resistance.

On the other side, the objective of the study on the molecular dataset is assessing and predicting the value of the docking energy of genistein based drug compounds with estrogen receptor proteins.

Results of this two studies are illustrated into (Archetti et al., 2009a) and (Archetti et al., 2009b) respectively. A description of the dataset used is in Appendix A.

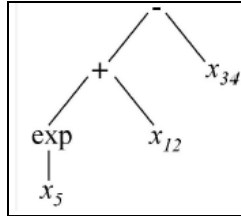
In both regression problems, we have used a tree-based GP configuration inspired by (Koza, 1992). Each feature in the dataset has been represented as a floating point number. Potential solutions (GP individuals) have been built by means of the set of functions  $F = \{+, -, /, \log, \sin, \cos, \exp, \text{sqrt}\}$ .

The set of terminals  $T$  was composed by  $M-1$  floating point variables (where  $M$  is the number of features in the dataset). The fitness function we have used is the Root Mean Squared Error (RMSE) on the training set.

Below is reported an example of this kind of function: suppose the names of the floating point variables contained into the tree  $T$  set are  $x_1, x_2, \dots, x_{M-1}$ . Then, a candidate solution found by GP, expressed in infix notation, may for instance be:

$$F(x_1, x_2, \dots, x_{M-1}) = \exp(x_5) + x_{12} - x_{34} \quad (2-1)$$

A tree representation of expression ( 2-1) is given in Fig. 2-2.



**Fig. 2-2:** An example of a simple GP individual

Its fitness is obtained by calculating the value of  $F(x_1, x_2, \dots, x_{M-1})$  on all the lines of the dataset, i.e. by assigning the values contained in each line of the dataset to variables  $x_1, x_2, \dots, x_{M-1}$  (in this example, only three variables are used). The RMSE is calculated using this result and the corresponding target for each line of the training set.

In order to improve GP performances, we have optimized the RMSE on the training set with linear scaling, as described in (Keijzer, 2004). The efficacy of linear scaling in GP for many regression problems has been shown in (Keijzer, 2003; Keijzer, 2004). In order to give a picture of the generalization ability of GP, we also report the RMSE and the correlation coefficient (CC) between outputs and targets on the test set.

The parameters used in the implemented GP experiments are reported in Tab. 2-1. Furthermore, we have used generational tree based GP with elitism, i.e. unchanged copy of the best individual into the next population at each generation. Finally, no explicit feature selection strategy has been employed (data from our dataset have been used as input to GP with no filtering, nor pre-processing) in the experiments presented here, since we want to point out GP's ability to automatically perform an implicit feature selection.

<i>Parameter</i>	<i>Value</i>
Population size	200 individuals
Population initialization	Ramped half and half
Selection method	Tournament
Tournament size	7
Genetic operators	Sub tree crossover and mutation
Crossover rate	0.95
Mutation rate	0.1
Maximum number of generation	500

**Tab. 2-1:** Parameters used in the presented GP experiments

### *Genetic Programming for classification*

The aim of this application is not only to underline the ability of GP to perform a feature selection, but also to classify tumour tissues belonging to the "Colon dataset" and "Leukemia dataset", both presented in appendix A.

Also in this case the representation of individuals, that will be candidate classifiers are Lisp-like tree expressions built using the function set  $F = \{+, *, -, /\}$  and a terminal set  $T$  composed by  $M$  floating point variables, where  $M$  is the number of columns in the dataset (i.e.,  $M = 2000$  for the Colon Dataset and  $M = 7070$  for the Leukemia Dataset). Thus, GP individuals are arithmetic expressions that can be transformed into binary classifiers (class "normal" for healthy tissues and class "tumour" for ill ones for the Colon Dataset; class "myeloid" for acute myeloid leukemia and class "lymphoblastic" for acute lymphoblastic leukemia for

the Leukemia Dataset) by using a threshold. Here, we use two fitness functions: *ROCAUC* and *CCI*. In the first case each classifier is evaluated by a fitness function defined as the area under the ROC curve (Metz, 1978; Zweig and Campbell, 1993). In this case, the ROC curve is obtained by considering 20 different threshold values uniformly distributed in the interval  $[-1, 1]$ . For each one of these threshold values, a point is drawn having as abscissa the false positive rate and as ordinate the true positive rate obtained by the candidate classifier using that threshold. The area is calculated using the trapezoids method. The second type of fitness function is instead obtained by fixing a particular threshold value (equal to 0.5 in this work, following (Roskopf et al., 2007)) and calculating the CCI. CCI is defined as the correctly classify instances rate, that is,  $CCI = (TP + TN)/N$ , where *TP* indicates True Positives, *TN* specifies True Negatives, and *N* is the number of rows in the dataset. For calculating both these fitness values during the presented GP simulations, we have considered a static and a dynamic way of handling the training set, and we have considered training data as they are (i.e., without any explicit modification) or perturbing them with noise.

### 2.1.3.2 Experimental Results on NCI60 dataset

As just said before, our goal is two folds: find a mathematical relationship between the gene expression profile and the activity pattern of some particular drugs and test the ability of GP as a features selection technique.

We consider four particular drugs, chosen from the NCI-60 A-matrix: Fluorouracil (5-FU), Fludarabine, Floxuridine and Cytarabine. For each one of these drugs we have built a separate dataset. Each one of our four datasets can be represented by  $N \times M$  matrices  $H = [H(i,j)]$  where  $N = 60$  and  $M = 1376$ . Each line *i* represents a gene expression whose known value of the therapeutic response to the chosen drug (Fluorouracil, Fludarabine, Floxuridine and Cytarabine respectively) has been placed at position  $H(i,1376)$ . Thus, the last column of matrix *H* contains the known values of the parameter to estimate. The four matrices representing the dataset of each drug differ only in the last column, while all the other columns are genes expressions as contained in the NCI-60 T-matrix (see Appendix A.1.2).

The results that we have obtained with Linear Regression are shown in Tab. 2-2. These table is partitioned into three sub-tables, respectively reporting the results obtained with no feature selection, i.e. where data have been used with no filtering or pre-processing (upper table), with PCFS (table in the middle) and with CorrFS (lower table). For each one of these three sub-tables, the first column reports the name of the drug that has been used, the second column reports the Root Mean Squared Error (RMSE) on the test set and the third column reports the correlation coefficient (CC) between outputs and targets.

PCFS has selected 47 of the 1375 available features for all drugs, while CorrFS has selected 20 features for Fluorouracil, 19 features for Floxuridine, 30 features for Fludarabine and 25 features for Cytarabine.

**a)No Feature Selection**

Drug	RMSE on test set	CC on test set
Fluorouracil	0.2341	0.4815
Floxuridine	0.3227	0.2196
Fludarabine	0.1811	0.5927
Cytarabine	0.3391	0.4761

**b)Principal Component Based Feature Selection (PCFS)**

Drug	RMSE on test set	CC on test set
Fluorouracil	0.4946	0.6159
Floxuridine	0.5807	0.2164
Fludarabine	0.4910	0.1353
Cytarabine	0.7004	0.0693

**c)Correlation Based Feature Selection (CorrFS)**

Drug	RMSE on test set	CC on test set
Fluorouracil	0.1433	0.8675
Floxuridine	0.1642	0.6828
Fludarabine	0.2104	0.6979
Cytarabine	0.3370	0.5252

**Tab. 2-2:** Experimental results returned by Linear Regression for therapeutic responses prediction of four different drugs.

In Tab. 2-3 we report the results returned by Least Square Regression on the same datasets. For obtaining these results, we have performed 100 independent runs of the Weka implementation of Least Square Regression (Weka, 2006) and we report the best, the average and the standard deviations of the results obtained.

As for the case of Linear Regression, we have applied the method with no feature selection (part (a) of the table) and using the PCFS (part (b)) and CorrFS (part (c)) feature selection methods.

Comparing the results of Tab. 2-2 with the ones of Tab. 2-3, we observe that for Floxuridine, both the best RMSE and the best CC have been obtained by Linear Regression using CorrFS. On the other hand, for Fluorouracil, Fludarabine and Cytarabine both the best RMSE and the best CC have been obtained by Least Square Regression using CorrFS. However, we point out that average results obtained by Least Square Regression are comparable (and sometimes slightly worse) than the ones returned by Linear Regression. Standard deviations for the Least Square Regression method are rather small and this seems to hint the fact that the behaviour of this method is rather “stable” (different executions return results which are rather “similar” to each other). Furthermore, we observe that the PCFS method is not useful to improve results on Tab. 2-4 reports the results we have obtained executing 100 independent GP runs with no previous explicit feature selection.

a)No Feature Selection				b)Principal Component Based Feature Selection (PCFS)			
Drug		RMSE on test set	CC on test set	Drug		RMSE on test set	CC on test set
Fluorouracil	Best	0.2446	0.3505	Fluorouracil	Best	0.3183	0.5761
	Avg	0.2591	0.2077		Avg	0.3275	0.4160
	Std. Dev.	0.0160	0.0160		Std. Dev.	0.0161	0.0139
Floxuridine	Best	0.3518	0.1695	Floxuridine	Best	0.4110	0.2142
	Avg	0.3595	0.2836		Avg	0.4208	0.1783
	Std. Dev.	0.0351	0.0169		Std. Dev.	0.0136	0.0420
Fludarabine	Best	0.1961	0.5207	Fludarabine	Best	0.3263	0.1885
	Avg	0.2098	0.4591		Avg	0.3501	0.1056
	Std. Dev.	0.0106	0.0469		Std. Dev.	0.0405	0.0567
Cytarabine	Best	0.3517	0.0727	Cytarabine	Best	0.4435	0.1260
	Avg	0.3799	0.1524		Avg	0.5829	0.0497
	Std. Dev.	0.0189	0.0145		Std. Dev.	0.0103	0.0552

c)Correlation Based Feature Selection (CorrFS)			
Drug		RMSE on test set	CC on test set
Fluorouracil	Best	0.1389	0.8980
	Avg	0.1843	0.7352
	Std. Dev.	0.0481	0.0184
Floxuridine	Best	0.2009	0.7439
	Avg	0.2455	0.6071
	Std. Dev.	0.0419	0.0105
Fludarabine	Best	0.1661	0.7439
	Avg	0.2038	0.6071
	Std. Dev.	0.0338	0.0105
Cytarabine	Best	0.2902	0.7142
	Avg	0.3269	0.5616
	Std. Dev.	0.0376	0.0192

**Tab. 2-3:** Experimental results returned by Least Square Regression for therapeutic responses of four different drugs.

For each one of these runs, we have monitored the best RMSE and CC in the GP population and we have reported their best, average and standard deviations. In general, the best results found by GP have a slightly better RMSE and remarkably better CC compared to both Linear Regression and Least Square Regression for all the four considered drugs.

Tab. 2-4 also shows that standard deviations of the best obtained results over the 100 independent runs that we have executed are rather “small”, thus we could informally say that GP behaviour is rather “stable”, i.e. the results obtained in the different runs are quantitatively rather “similar” to each other.

**a) Individual with the best RMSE on the test set**

Drug		RMSE	CC	RMSE	CC
		on training set	on training set	on test set	on test set
Fluorouracil	Best	0.1426	0.9348	0.1126	0.9006
	Avg	0.1362	0.8961	0.1687	0.7526
	Std. Dev.	0.0203	0.0576	0.0151	0.1803
Floxuridine	Best	0.1968	0.8056	0.1225	0.8628
	Avg	0.1968	0.8056	0.1225	0.8628
	Std. Dev.	0.0259	0.1217	0.0099	0.1902
Fludarabine	Best	0.1201	0.9055	0.1065	0.9675
	Avg	0.1580	0.8207	0.1544	0.7065
	Std. Dev.	0.02079	0.0963	0.0155	0.2399
Cytarabine	Best	0.1813	0.6694	0.1967	0.8815
	Avg	0.2370	0.6524	0.2601	0.6371
	Std. Dev.	0.0334	0.1457	0.0238	0.1378

**b) Individual with the best CC on the test set**

Drug		RMSE	CC	RMSE	CC
		on training set	on training set	on test set	on test set
Fluorouracil	Best	0.1112	0.9542	0.1487	0.9522
	Avg	0.1236	0.9299	0.1940	0.9046
	Std. Dev.	0.0171	0.0372	0.0208	0.0238
Floxuridine	Best	0.1860	0.08221	0.1347	0.9110
	Avg	0.1823	0.7962	0.1736	0.8127
	Std. Dev.	0.0253	0.0943	0.0175	0.0473
Fludarabine	Best	0.1204	0.9248	0.1089	0.9729
	Avg	0.1312	0.9060	0.1727	0.9113
	Std. Dev.	0.0162	0.0356	0.0216	0.0311
Cytarabine	Best	0.1767	0.8973	0.2003	0.8911
	Avg	0.2181	0.7469	0.2843	0.7253
	Std. Dev.	0.0321	0.1264	0.0367	0.0577

**Tab. 2-4:** Results returned by GP

Furthermore, we did not apply any explicit feature selection method before running GP, thus saving some computational effort.

We hypothesize that one of the advantages of GP compared to Linear Regression is that, while Linear Regression works under the hypothesis that the target function is linear, GP makes no hypothesis on the shape of the target function. Furthermore, the fact that GP allows us to obtain better CC results than Linear Regression and Least Square Regression may be due to the fact that we have used linear scaling as a quality criterion on the training set for GP: linear scaling is in fact known to optimize CC and RMSE together, as explained in (Keijzer, 2004).

We report the best individuals found by GP for the four considered drugs, in order to pinpoint the implicit ability of selecting relevant features and to understand the mutual relationships between genes that could potentially support the identification of biological meaningful pathways. The individual with the best RMSE found by GP over our 100 simulations for Fluorouracil, expressed in infix notation, is:

```
(id_376146+id_49816+div(id_361815, id_158260)+ div(id_376146,
(id_242740 + id_306136 + id_471110 * id_116296)) + id_428733 +
id_376146 + id_306136 + sin(id_346396) + id_43555)
```



Here genes are given as references to entries of the NCI-60 T-Matrix and they are documented in (Nci60 dataset, Appendix A1.2). The fact that the number of features is 1375 and ids have reference number larger than 1375 does not have to be surprising, given that we have used exactly the same identifiers as in NCI60 in order to facilitate results interpretation.

It is possible to remark that the solution reported above uses only 12 of the 1375 possible features, thus GP has effectively performed an *automatic feature selection*. Furthermore, gene id\_376146 (Cyb561) appears in three different positions in this expression. As explained in (Srivastava et al., 1995), this gene was found to be highly expressed in colon cancer cell lines and T cell lymphomas. This seems to hint that GP is maintaining pertinent information into the population. The individual with the best CC found by GP is:

```
(id_470160 + exp(id_292082, id_321203) + div((exp(id_292082,
id_321203) + id_417226 + id_327435), exp(id_292082, id_321203))+
id_417226 + id_471096+ div((id_292082 + id_417226 + id_327435),
exp(id_417125, id_328234)) + exp(exp(sqrt(id_470160), id_193562),
(id_193562 + id_301416)) + exp(sqrt(id_470160), id_143985) +
id_471096 _ (id_292082 + id_417226) + exp(div(id_292082, id_359769),
id_471096) + exp((sqrt(id_470160) - id_327435), sqrt(id_470160)) +
id_488118) _ id_327435
```

Also in this case the individual uses a small set of features (16 out of 1375) and some features appear more than once; for instance, it is the case of gene id\_470160 (casp4) that appears in 5 different positions of this expression and of gene id\_292082 (ssr3) that appears in 6 different positions. Gene id\_470160 (casp4) encodes a protein that is a member of the cysteine-aspartic acidprotease (caspase) family and when over-expressed, it has been shown to induce cell apoptosis.

Gene id\_292082 (ssr3) is a glycosylated endoplasmic reticulum membrane receptor associated with protein translocation across the endoplasmic reticulum membrane and it is limited to cell-lines of leukemic origin.

Similar considerations can be done for the individuals with the best RMSE and CC found by GP for the other three drugs studied.

All these results underline the fact that GP has implicitly performed a feature selection for all the four drugs and that some genes and structures appear more than once in the expressions of the best solutions.

Since all these experiments are done on only one drug, a possible future work in order to include GP into the feature selection component of the clustering workflow depicted into Fig. 1-1, could be based on applying GP on all the set of drugs. After obtaining a single regression for each drug, we could apply a frequentist approach in order to rank the features selected based on their frequencies.

### 2.1.3.3 Experimental Results on Molecular Dataset

Principal goals of GP application on molecular dataset is to assess and predict the value of the docking energy of genistein based drug compounds with estrogen receptor proteins and to test with also in this case study the GP features selection ability.

Also in this case study different fitness function has been used:

- root mean squared error (RMSEGP);
- correlation coefficient (CCGP) between outputs and targets;
- RMSE with linear scaling (LinScalGP)

Tab. 2-5 reports the results of RMSEGP. These tables must be interpreted as follows: the upper part (part (a)) shows the results that we have obtained when no feature selection strategy has been employed (data from our dataset have been used as input with no filtering, nor pre-processing), the middle part (part (b)) reports the results when PCFS has been used and the lower part (part (c)) shows the results obtained using CorrFS. Columns 2 and 3 of these tables report the results obtained on the training set and columns 4 and 5 the ones on the test set; on both cases, we have reported the root mean squared error (RMSE) and the correlation coefficient (CC) between outputs and goals returned by the trained model.

These results have been obtained by executing 100 independent RMSEGP runs. For each one of these runs, we have monitored the individual with the best RMSE on the test set and the one with the best CC on the test set. The upper part of Tab. 2-5 reports the best (first line), average (second line) and standard deviation (third line) of the results returned by the individuals with the best RMSE on the test set at each run. The lower part of Tab. 2-5 does the same thing for the individuals with the best CC on the test set at each run.

<b>a) Individual with the best RMSE on the test set</b>				
	<i>RMSE</i>	<i>CC</i>	<i>RMSE</i>	<i>CC</i>
	on training set	on training set	on test set	on test set
Best	0.0805	0.7592	0.1104	0.7100
Avg	0.0899	0.7022	0.1227	0.6509
Std. Dev.	0.0056	0.0442	0.0059	0.0367
<b>b) Individual with the best CC on the test set</b>				
Best	0.0830	0.7913	0.1110	0.7323
Avg	0.0913	0.6924	0.1268	0.6659
Std. Dev.	0.0069	0.0532	0.0084	0.0330

**Tab. 2-5:** Results that we have obtained performing 100 independent runs of RMSEGP on our dataset.

Tab. 2-6 and Tab. 2-7 show the experimental results that have been returned by two non-evolutionary machine learning techniques: Linear Regression (Akaike, 1973) and Least Square Regression (Rousseeuw and Leroy, 1987).

**a) No Feature Selection**

	<i>RMSE</i> on training set	<i>CC</i> on training set	<i>RMSE</i> on test set	<i>CC</i> on test set
Best	0.0816	0.7265	0.1169	0.6952
Avg	0.0903	0.6754	0.1175	0.6432
Std. Dev.	0.0064	0.0467	0.0071	0.0391

**b) Principal Component Based Feature Selection (PCFS)**

Best	0.1054	0.6951	0.1328	0.6003
Avg	0.1183	0.6592	0.1395	0.5835
Std. Dev.	0.0082	0.0362	0.0052	0.0746

**c) Correlation Based Feature Selection (CorrFS)**

Best	0.0945	0.7064	0.1185	0.6972
Avg	0.0995	0.6845	0.1276	0.6325
Std. Dev.	0.0056	0.0476	0.0036	0.0427

**Tab. 2-6:** Experimental results returned by Linear Regression

**a) No Feature Selection**

	<i>RMSE</i> on training set	<i>CC</i> on training set	<i>RMSE</i> on test set	<i>CC</i> on test set
Best	0.0945	0.6964	0.1709	0.4145
Avg	0.1769	0.6065	0.1865	0.4395
Std. Dev.	0.0085	0.0392	0.0083	0.0426

**b) Principal Component Based Feature Selection (PCFS)**

Best	0.0971	0.6837	0.1805	0.4531
Avg	0.0996	0.6046	0.1965	0.4297
Std. Dev.	0.0047	0.0265	0.0085	0.0385

**c) Correlation Based Feature Selection (CorrFS)**

Best	0.0901	0.7013	0.1661	0.5143
Avg	0.0983	0.6954	0.1753	0.4975
Std. Dev.	0.0038	0.0238	0.0029	0.0285

**Tab. 2-7:** Experimental results returned by Least Square Regression

Tab. 2-8 reports the results of CCGP. This table must be interpreted as Tab. 2-5 and it clearly shows that if we optimize the correlation on the training set, we obtain a CC on the test set which is considerably better than the CC returned by RMSEGP and non-evolutionary techniques. Nevertheless, CCGP also returns poor RMSE results. We also point out that standard deviations on the RMSE are high, if compared with the ones obtained with RMSEGP, both on the training and test set.

These results suggest that only optimizing the correlation is not a good strategy to solve our problem. On the other hand, we would like to develop a method to optimize both the RMSE and the CC, and we hope that in that way we will be able to obtain results which are comparable to the ones of CCGP for the correlation, but better RMSE results. This is done with the next proposed configuration which takes both criteria, RMSE and the CC, into account.

**a) Individual with the best RMSE on the test set**

	<i>RMSE</i> on training set	<i>CC</i> on training set	<i>RMSE</i> on test set	<i>CC</i> on test set
Best	0.0893	0.8017	0.1225	0.6618
Avg	0.1202	0.8329	0.1446	0.5600
Std. Dev.	0.0190	0.0420	0.0094	0.0975

**b) Individual with the best CC on the test set**

Best	4.2266	0.9137	6.2392	0.9020
Avg	5.9515	0.9070	7.6128	0.8758
Std. Dev.	4.0070	0.0093	5.0734	0.0195

**Tab. 2-8:** Results obtained with CCGP configuration

Moreover we have decided to use linear scaling, like we have adopted in the NCI60 dataset case study.

Tab. 2-9 clearly shows that both the best RMSE and CC on the test set found by LinScalGP are better than the best RMSE and CC found by the other GP variants. Furthermore, also the average best RMSE and the average best CC outperform the best RMSE and CC found by any of the other techniques. Finally, standard deviations confirm that the behaviour of LinScalGP is “stable” (i.e. the results of the 100 runs are rather similar to each other). All these considerations allow us to conclude that LinScalGP seems a suitable technique to solve our problem.

**a) Individual with the best RMSE on the test set**

	<i>RMSE</i> on training set	<i>CC</i> on training set	<i>RMSE</i> on test set	<i>CC</i> on test set
Best	0.0740	0.9193	0.1000	0.9065
Avg	0.0757	0.8939	0.1092	0.8781
Std. Dev.	0.0055	0.0180	0.0036	0.539

**b) Individual with the best CC on the test set**

Best	0.0691	0.9356	0.1000	0.9245
Avg	0.0735	0.9221	0.1107	0.9057
Std. Dev.	0.0041	0.0113	0.0042	0.0074

**Tab. 2-9:** Results obtained with linScalGP configuration

The genotypes of the individual with the best RMSE will be given here as expressions in infix form and the molecular descriptors will be represented using traditional identifier.

```
(POLA_pmi + (SMR_SAS0 - Z_pcplus + VAdjMa) * (b_lrotR + chi0v_C +
POLA_pmi + b_lrotR + SlogP_VOL0) * (SMR_SAS0 - Z_pcplus)*(chi0v +
VAdjMa)) * (chi0v + VAdjMa) + chi0v_C +(SMR_SAS0 - Z_pcplus)*(chi0v +
VAdjMa) + VAdjMa
```

The first thing that one might observe when looking at this expression is that it uses a limited number of molecular descriptors: only 12 different descriptors over the 267 total descriptors included in the dataset. In other words, although no explicit feature selection algorithm has been applied to reduce the number of input data, *GP has implicitly performed a strong feature selection.*

The mechanism that allows GP to perform feature selection is simple: GP searches over the space of all arithmetic expressions of 267 variables. This search space includes the expressions that use all the 267 variables, but also the ones that use a smaller number of variables and in principle there is no reason why an expression using a smaller number of variables could not have a better

fitness value than an expression using all the 267 variables. If expressions using smaller number of variables get a better fitness, they survive, given that fitness is the only principle used by GP for selecting genes. This is evidently what happened during the presented GP executions: GP has found expressions using a small number of variables with a better fitness value than the ones using all variables. Thus, the former expressions survived into the population, while the latter ones were extinguished.

Furthermore, we point out that all the descriptors that have been used have an intuitive correlation with docking energy. In fact they are mostly belonging to the categories of constitutional descriptors, derived from properties like solvent accessible surface and log P, characteristics known to influence the binding energy.

#### **2.1.3.4 Experimental Results on Oncological Datasets**

In this study, we present an application of Genetic Programming for molecular classification of cancer and for the identification of the principal genes that explained the studied pathologies.

Four versions of GP are studied on those datasets; those GP variants differ by the way of handling the training set and by the fact that they may or may not affect training data with noise.

Results returned by GP are compared with the ones returned by three well-known non-evolutionary Machine Learning methods: Support Vector Machines, Multi-Boosting and Random Forests.

Combining different methods of handling training set and data have lead us to define four different versions of GP, that we call GP0, GP1, GP2, and GP3 for simplicity.

- *GP0* uses the static training set handling and data with no noise. This corresponds to *standard* GP.
- *GP1* uses the static training set handling and data perturbed with Gaussian noise.
- *GP2* uses the dynamic training set handling and data with no noise.
- *GP3* uses the dynamic training set handling and data perturbed with Gaussian noise.

Results obtained by the non-evolutionary methods and by the different GP variants on the Colon Dataset and on the Leukemia Dataset are reported in the sequent paragraphs.

##### *Colon dataset*

Tab. 2-10 summarizes the experimental results obtained by the non-evolutionary methods on the Colon Dataset. SVM is the method that returns the best average results, both for CCI and ROC, while the best CCI results are returned by Random Forests and SVM, and the best ROC results are returned by Random Forests. We point out that we have applied these classification methods to our datasets without any explicit feature selection or pre-processing algorithm. The motivation for this is that we wanted to compare these results with the ones obtained by GP, pointing out that GP is able to perform an

automatic feature selection, while the other non-evolutionary methods do not have this capability.

	CCI			ROC		
	Best	Average	Std. Dev	Best	Average	Std.Dev
Random Forests	0.9444	0.7417	0.0810	1	0.8250	0.0755
SVM	0.9444	0.8778	0.0438	0.9545	0.8525	0.0874
Multi Boosting	0.8889	0.7850	0.0577	0.9861	0.8152	0.0488

**Tab. 2-10:** Results returned by non- evolutionary methods on Colon dataset

	CCI			ROC		
	Best	Average	Std. Dev	Best	Average	Std.Dev
GP0	1	0.8926	0.038	1	0.9437	0.0472
GP1	1	0.8946	0.042	1	0.9444	0.0455
GP2	1	0.8947	0.039	1	0.9437	0.0455
GP3	1	0.8950	0.042	1	0.9555	0.0466

**Tab. 2-11:** Results returned by the studied GP variants on the Colon dataset

Tab. 2-11 reports the results obtained by the different GP variants studied using the same 10 training-test partitions as in Tab. 2-10. Comparing the results reported in these two tables, we can remark that all GP variants are able to find an ideal solution both for CCI and ROC, which is not the case for the non-evolutionary methods (with the exception of Random Trees for ROC). Also comparing the average values, we can remark that all GP variants outperform all non-evolutionary methods, and the respective standard deviations seem to hint that the difference between GP performances and the ones of the other methods is statistically relevant.

Differences between the various GP variants seem marginal, which hints that both the dynamic dataset handling and the use of Gaussian noise are not useful to improve GP generalization ability, at least for this application. By the way, it has to be remarked that performances of standard GP (GP0) are already (informally) rather “high”, and thus difficult to improve. In the future, we plan to investigate the gain in using GP1, GP2, and GP3 for more complex problems, where GP0 is not able to find good solutions.

*Leukemia dataset*

	CCI			ROC		
	Best	Average	Std. Dev	Best	Average	Std.Dev
Random Forests	0.9048	0.7191	0.0939	0.9500	0.6999	0.1270
SVM	0.8571	0.7476	0.0552	0.8375	0.7274	0.0924
Multi Boosting	0.9524	0.7548	0.0733	1	0.7500	0.0895

**Tab. 2-12:** Results returned by the non evolutionary methods on the Leukemia dataset

	CCI			ROC		
	Best	Average	Std. Dev	Best	Average	Std.Dev
GP0	1	0.8323	0.0390	1	0.8491	0.0047
GP1	1	0.8592	0.0425	1	0.8777	0.0400
GP2	1	0.8325	0.0395	0.9778	0.8500	0.0392
GP3	1	0.8607	0.0407	0.9904	0.8778	0.0381

**Tab. 2-13:** Results returned by the studied GP variants on the Leukemia dataset

Results obtained by the studied non-evolutionary methods are summarized Tab. 2-12. For the Leukemia Dataset, MultiBoosting is the method that has returned both the best results and the best average results, both for CCI and ROC.

Tab. 2-13 reports the results obtained by the different GP variants studied using the same 10 training-test partitions as in Tab. 2-12. Also in this case, all GP variants outperform all non-evolutionary methods, and standard deviation values seem to hint that the differences between the average results obtained by GP and the average ones obtained by the best non-evolutionary method on this dataset (Multi Boosting) are statistically relevant. All GP variants have been able to produce ideal solutions for CCI, while only GP0 and GP1 have been able to generate ideal ROC values. We finally remark that, also for the Leukemia Dataset, perturbing data with Gaussian noise or handling the training set in a dynamic way is not beneficial.

We now report the genotype of some of the best solutions found by GP in the form of expressions in infix notation, and successively we describe the most recurrent genes contained in them.

These expressions are reported here to allow the reader to have an idea of how the best solutions found by GP on the test sets look like; we do not pretend them to necessarily be the model explaining the relationships between gene expressions and the studied pathologies. In order to build such a model, collaborations with domain experts are needed (and we are planning them in our future activity). Nevertheless, we hope that reporting those expressions here may be a starting point for this new and challenging research. Furthermore, we also report scatter plots of the Z-scores of the different genes contained in the best solutions found by GP, and we show how those values are correlated when ROC and CCI are used as fitness functions.

*Colon Dataset*

We first report a solution with CCI = 1 on the test set found by GP0.

```
IF (K03460%X59131*(X66924 + H20709) - (T74896 + U28963)*(R61359 +
    T86444) - (U20659 - T81460)*R53941)>0.5
THEN Class = "tumour"
ELSE Class = "normal"
```

We remark that GP has performed an automatic feature selection; in fact, this solution contains only 15 over the 2000 possible genes. This fact distinguishes GP from the other studied Machine Learning, which can use a subset of features only if an explicit feature selection algorithm is executed before training (pre-processing).

One of the solutions with area under the ROC curve on the test set equal to 1 returned by GP0 is

```
IF ((X51416+R99200 * X06614)%(H23544 * X61123-T47213+M34344+(H79575-
    R50864) * U18920 + R46739 %(U20659 + H04333)-R53941+L09604)>0.5)
THEN Class = "tumour"
ELSE Class = "normal"
```

In this case, GP’s feature selection has been even stronger: only 11 of the 2000 available genes are used by GP.

It is a widely agreed upon idea that only a restricted number of genes are correlated with tumour pathologies (those genes are often identified by domain experts as biomarkers). For this reason, the ability of GP to retain a limited number of genes into the proposed solutions is interesting. In order to identify and study the most important genes found by GP, for each one of the 4000 GP independent runs that we have performed to obtain the results reported in this Thesis (100 independent runs for each one of the 10 training-test different partitions and for each one of the 4 GP variants), we have retained the best solution found on the test set, both for CCI and ROC. In all those 8000 solutions, we have counted the number of occurrences of each gene in the dataset. We finally have extracted the 30 most recurrent genes. A detailed description of those genes is contained in (Archetti et al., 2009b).

Furthermore, we have considered all the genes that have appeared in at least one best solution found by GP using CCI and in at least one best solution found by GP using ROC (i.e., we have considered the set of genes contained in the best solutions found by GP using CCI, set of genes contained in the best solutions found by GP using ROC, and we have considered the intersection between these two sets).

In Fig. 2-3 the normalized Z-Score of these genes is depicted. In particular, gene’s normalized Z-Score it is defined as follows:

$$Z - Score = \frac{S_i - E(S_i)}{\sigma} \tag{2-2}$$

where  $S_i$  denotes the number of times genes  $i$  being contained in the studied GP solutions,  $E(S_i)$  is the expected number of times for gene  $i$  being contained in those solutions, and  $\sigma$  denotes the square root of the variance.



The computation of  $E(S_i)$  is:

$$E(S_i) = \frac{\text{number of gene contained in the studied GP solution}}{\text{number of genes in the initial gene pool}} \quad (2-3)$$

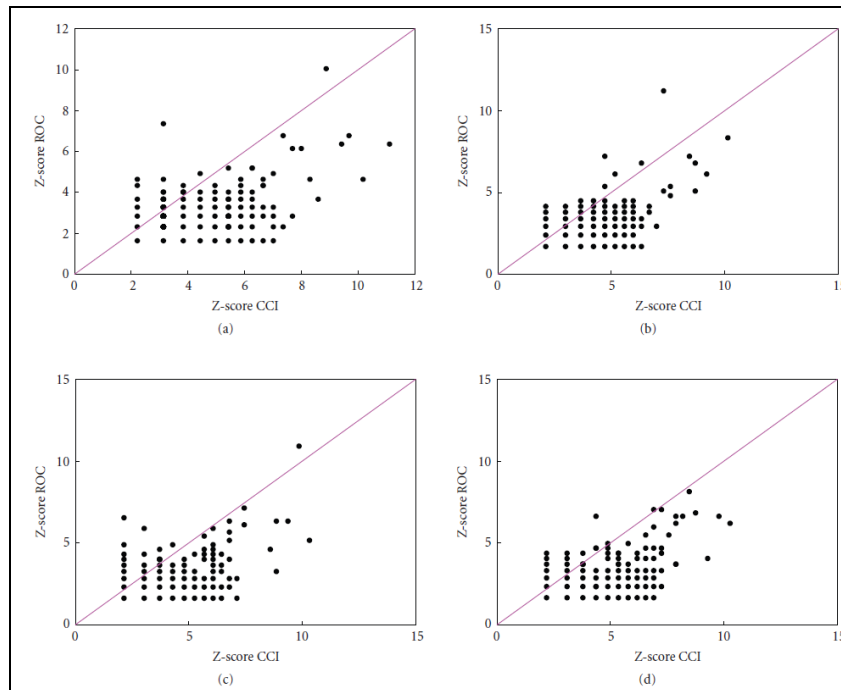


Fig. 2-3: Normalized Z-score of the most recurrent common genes for the Colon dataset.

In particular, Fig. 2-3 shows the correlation between genes' normalized Z-Score for the two fitness criteria (CCI and ROC) for the four versions of GP ((a): GP0, (b): GP1, (c): GP2, (d): GP3) that we have studied. For all these GP versions, the score seem positively correlated (the figure also reports the axis bisector, which represents the ideal correlation).

### Leukemia dataset

The genotype of one of the solutions with CCI = 1 found by GP0 is:

```
IF (X05409 % M28130 + (U94855 - M84526) % (U04270* X55668 % D28473-
(D38498 - Z37976) % M96326) > 0.5)
THEN Class = "tumour"
ELSE Class = "normal"
```

Also in this case, GP has operated an automatic feature selection, given that this solution contains only 10 of the 7070 possible genes.

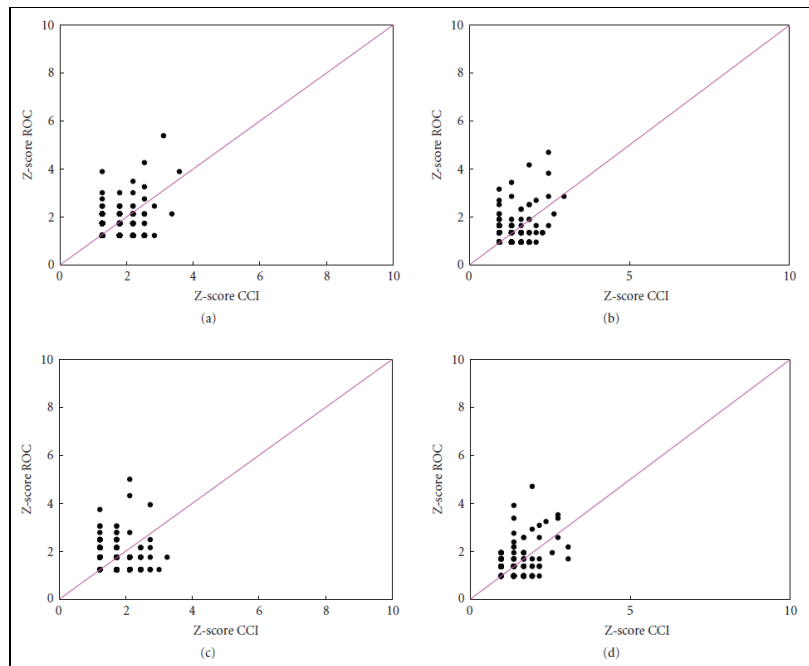
The genotype of a solution with area under the ROC curve on the test set equal to 1 returned by GP0 is:

```
IF (U15782 - J04990) % X04707+ X62822 - M27891*M96326 > 0.5)
THEN Class = "tumour"
ELSE Class = "normal"
```

It contains only 6 of the 7070 possible genes.

Also for the Leukemia dataset for each one of the 4000 GP independent runs, we have retained the best solutions found on the test set, both for CCI and ROC. In all those 8000 solutions, we have counted the number of occurrences of each gene in the dataset. We finally have extracted the 30 most recurrent genes. For a detailed description of those genes we remand the reader to (Archetti et al., 2009b).

In Fig. 2-4 we report the correlation between the normalized Z-Scores of the genes that appear at least once in the best solutions found by GP using CCI and at least once in the best solutions found by GP using ROC. Also in this case, Z-Scores seem positively correlated.



**Fig. 2-4:** Normalized Z-score of the most recurrent common genes for the Leukemia dataset.

All the case studies presented in this subsection have confirmed that GP is a promising techniques for automatically perform, given a dataset as input, a feature selection. In the last case a study regarding the importance of the selected features has been presented, in order to confirm the ability of GP. So in this way we could include GP in the feature selection component of clustering procedures presented in Fig. 1-1.

## 2.2 Mixed Data Types

The clustering methods we are been considering in section 1, while very good at grouping numerical values i.e. gene expression levels, cannot incorporate phenotypic data about the samples like histopathology observations (generally categorical values).

In the last years the data mining community has been on the look-out for good criterion function for handling mixed data, since the traditional clustering algorithms work well on either categorical or numeric valued data.

In order to overcome this problem, different strategies have been utilized like the follows:

- A simple approach in which categorical and nominal attribute values are converted to numeric integer values. Subsequently numeric distance measures are used for calculating similarity between object pairs. On the other hand, it is very difficult to give correct numeric values to categorical values like colour, etc.
- Another approach has been to discretize numeric attributes and then apply a categorical clustering algorithm. Also in this case there is a drawback: the discretization process leads to loss of information.

In the next sections we report an overview of existing algorithms and we focalize our attention on a particular algorithm, called Modified-K-Prototypes that we will use as a traditional technique in a particular application described in subsection 5.3.

### 2.2.1 Overview of Existing Algorithm

Clustering techniques for mixed data requires an objective function (that we already defined in section 1.2) with a combination of the distance measure for numerical values and that for categorical values. A simple example can be seen in (Bushel et al., 2007), where the sum of the distance for the numerical values and a matching distance for a categorical values has been computed.

(Li and Biswas, 2002) presented the Similarity Based Agglomerative Clustering (SBAC) algorithm based on (Goodall, 1966) similarity measure. This algorithm works well with mixed numeric and categorical features, though is computationally expensive.

(Huang, 1997) proposed an objective function that considers numeric and categorical attributes separately. This function handles mixed data sets and computes the similarity between two elements in terms of two distance values (one for numeric attributes and the other for categorical attributes) and since it can be used with a partitional algorithm, is cost-effective. In particular, (Huang, 1997) defined an objective function, that must be minimized, for clustering mixed data sets with  $n$  data objects and  $m$  attributes ( $m_r$  numeric attributes,  $m_c$  categorical attributes,  $m = m_r + m_c$ ) as

$$\zeta = \sum_{i=1}^n d(x_i, q_j) \quad (2-4)$$

where the distance  $d(x_i, q_j)$  of a data object  $x_i$  from the closest cluster centroid  $q_j$  is defined as:

$$d(x_i, q_j) = \sum_{t=1}^{m_r} (x_{it}^r - q_{jt}^r)^2 + \gamma_j \sum_{t=1}^{m_c} \delta(x_{it}^c - q_{jt}^c) \quad (2-5)$$

Where:

- $x_{it}^r$  are values of numeric attributes and  $x_{it}^c$  are values of categorical attributes for data object  $x_i$ .
- $q_j = (q_{j1}, q_{j2}, \dots, q_{jm})$  represents the cluster centroid for cluster  $j$ .
- $q_{jt}^c$  represents the most common value (mode) for categorical attributes  $t$  and class  $j$ . For these attributes,  $d(p, q) = 0$  for  $p = q$  and  $d(p, q) = 1$  for  $p \neq q$ .
- $q_{jt}^r$  represents the mean of numeric attribute  $t$  and cluster  $j$ .
- $\gamma_j$  is a weight for categorical attributes for cluster  $j$ .

Objective function  $\zeta$  (reported in equation ( 2-4 )) is minimized for clustering mixed data sets. Analyzing Huang's function, we can see that it takes care of categorical attributes separately. However, this has a few shortcomings:

- ✓ For categorical attributes, the cluster centroid is represented by the mode of the cluster rather than the mean. While this allays the problem of finding the mean for categorical values, there is information loss since the true representation of the cluster is not obtained. Only one attribute value represents the cluster, even though there may be close seconds or thirds.
- ✓ Binary distance between two categorical attribute values  $p$  and  $q$  is taken as  $d(p, q) = 0$  for  $p = q$  and  $d(p, q) = 1$  for  $p \neq q$ . This does not reflect the real situation appropriately. (Stanfill and Waltz, 1986) suggested that for supervised learning though it is observed that  $d(p, q) = 0$  for  $p = q$ , but it is not necessarily true that  $d(p, q) = 1$  for  $p \neq q$ . According to them  $d(p, q)$  is mostly different for different attribute value pairs and depends on the relative frequencies of value pairs within a class. This works even for clustering since it is usually not one attribute that determines the clusters but rather a collection of attributes. Thus, during clustering, attribute value co-occurrences among different attributes should be considered to compute  $d(p, q)$ . The distance measure in that case can take care of significance of an attribute. (Ganti et al., 1999) uses a similar approach to derive clusters though it does not explicitly define  $d(p, q)$ .
- ✓ In Huang's objective function weight of all numeric attributes is taken to be 1. The weight of categorical attributes is a user-defined parameter  $\gamma_j$ . However, in a real data set all numeric attributes may not have the same effect on clustering. Incorrect user-given values of  $\gamma_j$  may also lead to inaccurate clustering.

Later, (He et al., 2005) extended their earlier algorithm for clustering categorical data called "Squeezed algorithm" (He et al., 2002), to cluster mixed data. In particular, they propose a divide-and-conqueror technique to solve mixed clustering problem. First, the original mixed dataset is divided into two sub-datasets: the pure categorical dataset and the pure numeric dataset. Next, existing well established clustering algorithms (k-mean for numerical data and k-mode for categorical data), designed for different types of dataset are employed to produce corresponding clusters. Last, the clustering results on the categorical and numeric dataset are combined as a categorical dataset on which the categorical data clustering algorithm is employed to get the final output. (He et al, 2005) contribution is to provide an algorithm framework for the mixed

attributes clustering problem, in which existing clustering algorithms can be integrated.

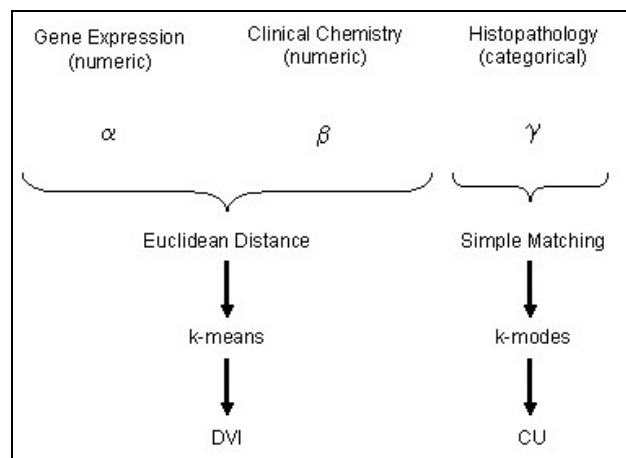
In the same year, (Huang et al., 2005) proposed a new “k-prototype clustering method” for handling mixed data. In this method attribute weights are automatically calculated based on the current partition of data. (Luo et al., 2006) proposed to cluster pure numeric subset of attributes and categorical attributes differently, and use cluster ensemble technique evidence accumulation to combine these clustering results to get final clusters.

In the last years, several other approaches have been proposed like those of (Andreopoulos, 2006), (Ahmad and Dey, 2007).

### 2.2.2 Modified K-prototypes Algorithm

In this section we present a particular algorithm for mixed data types, proposed by (Bushel et al., 2007), that we will subsequently use and modify for a peculiar life science application. The proposed algorithm consist in a modified k-prototypes, called “modk-prototypes”, algorithm.

The approach follows the k-means paradigm with randomization of initialization of the algorithm. The distance computation schemes for handling numeric and categorical values have been designed to take care of the shortcomings discussed above. The strategy involves constructing an objective function from the sum of the squared Euclidean distances for numeric data with simple matching for categorical values in order to measure dissimilarity of the samples. Separate weighting terms are used to control the influence of each data domain on the clustering of the samples. Finally, a dynamic validity index for numeric data was modified with a category utility measure in order to determine the optimal number of clusters in the mixed type data. A cluster's prototype is formed from the mean of the values for numeric features and the mode of the categorical values of all the samples in the group. The cluster's prototype is taken as a representation of the feature values that depicts the phenotype of the samples in the group.



**Fig. 2-5:** Schema of modk-prototypes (Bushel et al., 2007)

In Fig. 2-5 is represented the components of the modk-prototypes algorithm for mixed data types proposed by (Bushel et al., 2007).

The k-prototypes algorithm of (Huang, 2005) proposed above, was modified to

follow the k-means algorithm paradigm, and was also optimized to search for clusters formed closest to the global minima of the modk-prototypes objective function:

$$d(x_i, q_l) = \alpha \sum_{j=1}^{m_r} (x_{ij}^r - q_{lj}^r)^2 + \beta \sum_{j=1}^{m_s} (x_{ij}^s - q_{lj}^s)^2 + \gamma \sum_{j=1}^{m_c} (x_{ij}^c - q_{lj}^c) \quad (2-6)$$

where  $x_i$  is the  $i^{\text{th}}$  instance ( $i = 1 \dots N$ ),  $q_l$  is the  $l^{\text{th}}$  prototype, ( $l = 1 \dots k$ ),  $m_r$  is the number of numeric attributes, as for example microarray data,  $m_s$  is the number of numeric attributes belonging to another domain (as for example clinical chemistry),  $m_c$  is the number of categorical attributes.

Finally,  $\alpha$ ,  $\beta$  and  $\gamma$  denote the weights ( $W$ ) for the different data domain dissimilarity measures, respectively. They are non-negative and their sum is constrained to equal 1.

As we can see in Fig. 2-5, (Bushel et al., 2007) propose two different weights for numerical data and one for categorical data, since in his case study he applied Mod-k-prototype algorithm to two numerical data (microarray data and clinical chemistry data) and one categorical (histopathology data).

In particular, the weights for data domain  $t$  at the  $n^{\text{th}}$  step ( $W_t[n]$ ) are adapted (as follows:

$$W_t[n] = \begin{cases} \frac{1}{3} & n = 0 \\ (1 - \tau) \times W_t[n-1] + \tau \times \text{avercorr}(x^d, q^d) & \text{otherwise} \end{cases} \quad (2-7)$$

where  $\tau$  is the exponential weighting update factor in the range  $[0,1]$  and  $\text{avercorr}(x^d, q^d)$  is the average correlation coefficient (Pearson for numeric data, Jaccard for categorical data) between the samples and the prototypes based on the feature values from domain  $t$ .

$$\text{avercorr}(x^d, q^d) = \begin{cases} \left( \frac{1}{N} \sum_{i=1, X_i^d \in C_i}^N \left( \frac{\text{cov}(x_i^d, q_i^d)}{s_{x_i^d} s_{q_i^d}} \right) \right)^2 & \text{if domain } t \text{ is numeric} \\ \left( \frac{1}{N} \sum_{i=1, X_i^d \in C_i}^N \left( \frac{p_{(x_i^d, q_i^d)}}{p_{(x_i^d, q_i^d)} + 2q_{(x_i^d, q_i^d)}} \right) \right) & \text{if domain } t \text{ is categorical} \end{cases} \quad (2-8)$$

where  $\text{cov}$  is the instance covariance,  $s$  is the instance standard deviation,  $N$  is the number of instances,  $p$  is the number of features that match and  $q$  is the number of features that do not match.

#### Distance for numerical data

Letting  $z$  represent numerical data, the distance between  $x_i^z$  and  $q_l^z$  containing missing values is defined as:

$$d_j = \begin{cases} 0 & \text{if } x_{ij}^z \text{ or } q_{lj}^z \text{ is missing} \\ x_{ij}^z - q_{lj}^z & \text{otherwise} \end{cases} \quad (2-9)$$

Then the distance between  $x_i^z$  and  $q_i^z$  is:

$$d(x_i^z, q_i^z) = \frac{p}{p - p_0} \sum_{j=1}^p d_j^2 \quad (2-10)$$

where  $d_j$  is the Euclidean distance (described in subsection 1.1),  $p$  is the number of numeric features and  $p_0$  is the number of numeric features with missing values in  $x_i^z$  and  $q_i^z$  or both.

*Distance for categorical data*

For categorical ( $c$ ) feature values, the dissimilarity measure between  $x_i^c$  and  $q_i^c$  is defined by the total number of mismatches of the corresponding features from the instance  $x_i^c$  and the centroid  $q_i^c$  such that:

$$d(x_i^c, q_i^c) = \sum_{j=1}^{m_c} d(x_{ij}^c, q_{ij}^c) \quad (2-11)$$

Where

$$d(x_{ij}^c, q_{ij}^c) = \begin{cases} 0 & \text{if } x_{ij}^c = q_{ij}^c \\ 1 & \text{if } x_{ij}^c \neq q_{ij}^c \end{cases} \quad (2-12)$$

The mod $k$ -prototypes algorithm initialization is seeded by the domain data vector of a randomly selected instance for each of the  $k$  clusters. For adaptive clustering, recursion was used to update the centroids in order to find the configuration of the initial  $k$ -prototypes which ultimately results in the reduction of the objective function closest to the global minimum.

## 2.3 Knowledge “Integration”

### 2.3.1 Knowledge in Life Science Domain

During the second half of the 20<sup>th</sup> century life science domain and in particular, biology, has been dominated by a reductionist approach whose main enablers have been high throughput technologies, to generate large amounts of data, and bioinformatics to support their storage and analysis and help in inferring hidden relationships.

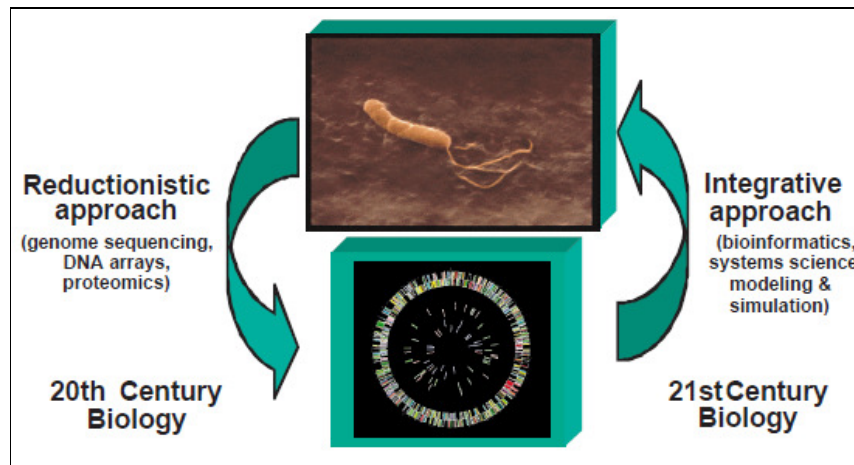


Fig. 2-6: Reductionist approach and Integrative approach.

We now have entire DNA sequences for a growing number of organisms and are continually defining their gene portfolios. Although functional assignment to these genes is presently incomplete, we can soon expect the assignment and verification of function for the majority of genes on selected genomes.

Expression array and proteomic technologies give us the capability to determine when a cell uses particular genes and when it does not.

The reductionist process is schematically depicted on the left in Fig. 2-6 (taken from (Palsson, 2000)). However, it has become generally accepted that the integrative analysis of the function of multiple gene products has become a critical issue for the future development of biology (Aebersold et al., 2000; Bailey, 1999; Evans, 2000; Hartwell, 1999; McAdams et al., 1998; Palsson 1997; Strothman, 1997).

Such integrative analysis will rely not only on bioinformatics, thru the development of semantically richer methods of analysis but increasingly on systems biology.

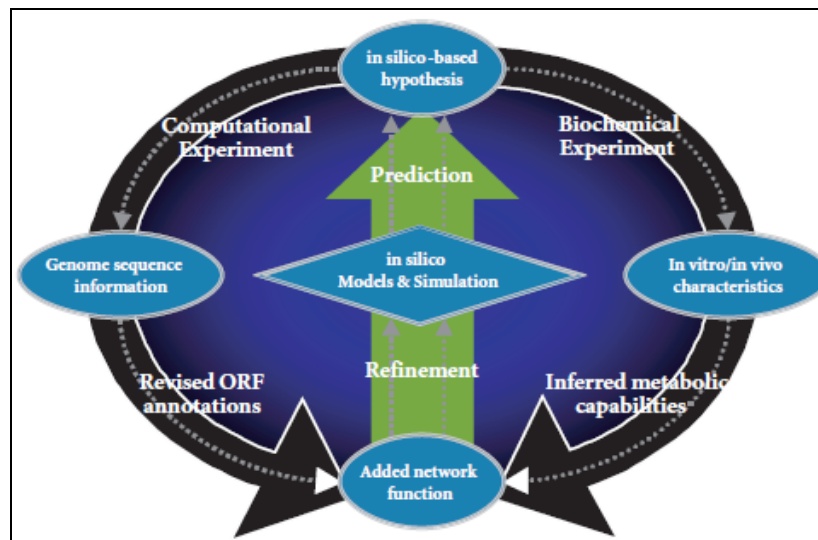
It is thus likely that over the coming years and decades, the biological sciences will be increasingly focused on the systems properties of cellular and tissue functions. These are the properties that arise from the whole, and represent “real” biological properties. These properties are sometimes referred to as “emergent” properties because they emerge from the whole and are not properties of the individual parts.



Bioinformatics will gradually merge into statistical and, both continuous and discrete, mathematical models: data driven iterative model building is likely to emerge as the underlying paradigm of integrative biology.

The process of building mathematical models of complex biological processes and their computer simulation will be an iterative one (visible in Fig. 2-7), for example beginning to construct “in silico organisms” that are computer representations of their in vivo counterparts. Initial versions will be synthesized using genomic, biochemical, and physiological data. These models will have some interpretive and predictive capabilities. However, because of incomplete knowledge of constraints and erroneous annotation, these initial models will be able to represent only some functions of the organism correctly.

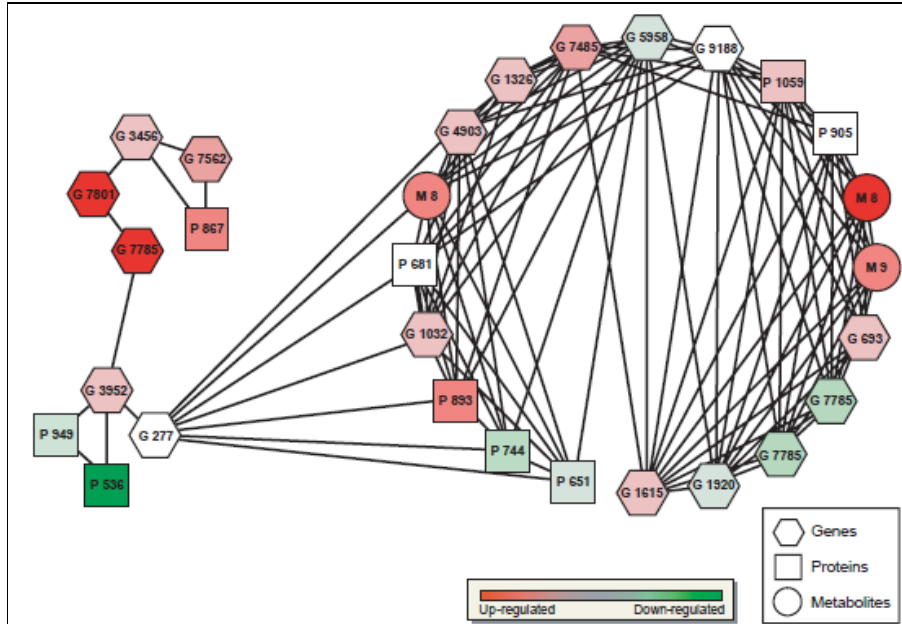
In carrying out this iterative model building process, we must learn to embrace failure. The main difference between the in silico and in vivo organism is that the in silico version is missing some features. Therefore, we must set out to formulate experimentally testable hypotheses based on the in silico analysis, perform the experiments, and update the models (see Fig. 2-7). Interestingly, this iterative process for building in silico organisms is likely to have two feedback loops. One is the classical experimental loop (the one on the right in Fig. 2-7), and the other is in silico (on the left in Fig. 2-7). Many corrections and adjustments for these models are likely to originate from analyzing and searching the ever-growing availability of bioinformatics databases.



**Fig. 2-7:** Iterative in silico model building in biology involves the formulation of experimentally testable hypotheses based in the in silico analysis, collection of experimental data, and subsequent refinement of the models based on these data.  
Figure from (Pallson, 2000)

It is clear that even though the molecular composition of living cells is complex (i.e. their genotype) the number of distinct behaviours (i.e. their phenotypes) that they display is much fewer. This important principle of simplicity from complexity is emerging from singular value decomposition of gene expression data that clearly shows that many expressed gene products behave in a highly coordinated fashion (Alter et al., 2000; Holter et al., 2000).

Nowhere knowledge driven bioinformatics and systems biology approaches are more needed than in the pharmaceutical industry: the reductionist ligand/receptor approach has been exposed as inadequate in the design of genomically base drugs and diagnostics: the very concept of target has changed from the single protein to the regulatory druggable network.



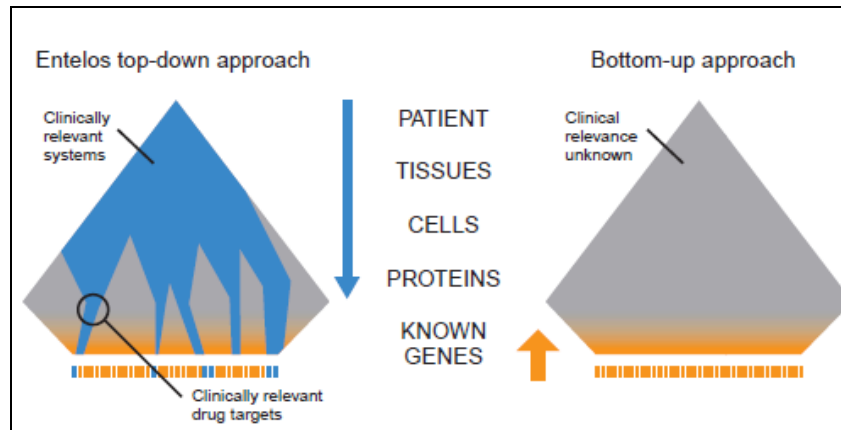
**Fig. 2-8:** Beyond Genomics Correlation Network

Up and down regulates genes, proteins and metabolites in a mammalian disease model versus controls ('normal'). A subset of the molecules depicted here could be used as an early biomarker of disease. The lines between and among the various molecules denote correlations, which are useful for understanding disease pathways and drug mechanisms of action from a biological perspective. Figure from (Mack, 2004)

For the pharma industry it is critical that bioinformatics and systems biology can move beyond data integration and use information from many data sets to create computational models that can predict phenotypes at the cell, tissue and organ level.

Inferring individual pathways is now possible: however no widely agreed upon methods still exist to infer the evolution of medically relevant phenotypes or clinical biomarkers, from molecular events in a cell.

Indeed it is increasingly clear that the sheer computational complexity of dealing in a principled way with all molecular components of a cell will prevent a purely bottom-up approach to allow the development of clinically useful disease models. For this reason bioinformatics, which has been so far mostly associated with data driven bottom-up approach must integrate itself with the top-down approach which is typical of systems biology ways (Fig. 2-9). In this figure, the analysis starts at the phenotype or even patient level and moves down thru the functional pathways towards protein networks and the underlying regulatory pathways.



**Fig. 2-9:** Top down versus bottom-up approaches.

Top-down approaches with major systems and work from the top down to the relevant tissues, cells, proteins and genes. Bottom-up starts with thousands of genes and proteins and tries to fit them together in a representation of cells.

### 2.3.2 Knowledge Integration in Clustering Procedure

In the introduction of this section we have explained and surveyed the importance of using the available background information in the process of building mathematical models to resolve important life science problems.

In fact, in many cases we have access to additional information or domain knowledge about the types of clusters that are sought in the data. This supplemental information may occur at the object level, such as class labels for a subset of objects, complementary information about “true” similarity between pairs of objects or about the relationships structure present in the data, or user preferences about how items should be grouped; or it may encode knowledge about the clusters themselves, such as their position, identity, minimum and maximum size, distribution.

Referring to the traditional clustering procedure described in section 1 and, in particular, in Fig. 1-1, we must modify this figure including different kind of background knowledge in the clustering problem.

The modified clustering procedure is visible in Fig. 2-10, where adaptations are highlighted in yellow. With the new dotted yellow components with yellow dotted, we want to include knowledge derivate from both clustering analysis and the supplemental information coming from external sources (for example for life science domain: annotation, Gene Ontology terms, information about experimental conditions, etc.).

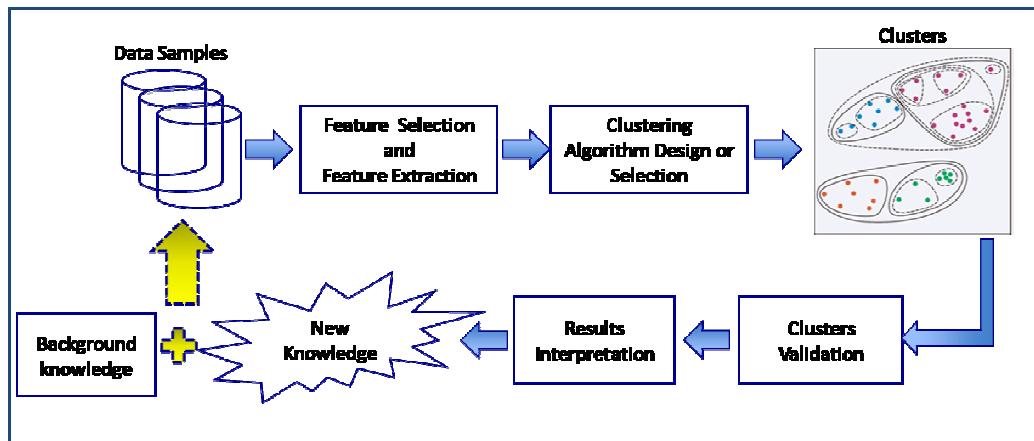


Fig. 2-10: Modified clustering procedure with knowledge integration

The “background” knowledge included into clustering algorithm can be used to guide it and consequently improve obtained results.

This knowledge takes into account not only a lot of information about initial data but also the richer structure of relationships present in data, more important for improving efficacy and efficiency of clustering output.

The addition of this information, and in particular of relational information, implies that in the traditional “flat” data representation (where each instance is represented by a vector with a fixed number of attribute (Duda et al., 2000)) used by traditional clustering approached, must include the rich relational structure. In fact in many real-world applications like biological data processing, social network analysis and text mining, data do not exist in isolation, but a rich structure of relationships subsists between data. A simple example can be viewed in biological domain, where there are al lot of relationships between genes and proteins based on many experimental conditions. Another example, maybe common, is the Web search domain where there are relation between documents and words in a text or web pages, search queries and web users.

Using all these relationships we can optimize the process of pattern discovery (clustering) between instances. As a result, *relational data clustering*, that learns cluster structures taking into account all these rich relationships structure, has become one of the most important data mining and machine learning topics.

Analyzing in a deep view relational data, we can view three major types of information:

- *attributes* for individual objects
- *homogeneous relations* between objects of the same type
- *heterogeneous relations* between objects of different types

For example, for a scientific publication relational data set of papers and authors, the personal information such as name, surname and affiliation for authors are attributes; the keywords relations among papers are homogeneous relations; the authorship relations between papers and authors are heterogeneous relations.

It's possible to find another simple example in clinical data set, where the personal patient's information is attributes, the concurrent medication among patients are homogeneous relations and the relations between patients and doctor through visits are heterogeneous.

The classic IID (independently and identically distributed) assumption in machine learning and statistics is violated by all these kind of data and the inclusion of all the different information about relationships in traditional machine learning methods represents a challenge.

The first instinctive solution is the transformation of relational data into a “flat” representation and subsequently the application of traditional clustering techniques on each type of objects independently. Few problems can arise from this kind of transformation like the loss of the relations and the rich structure of information. Secondly, in some data mining applications, users are not only interested in the hidden structure for each type of objects, but also interaction patterns involving multi-types of objects.

Furthermore, a large amount of clustering problems can be viewed as special cases of relational clustering.

For example, partitional clustering (like k-means algorithm explained in subsection 1.3.1) clusters homogeneous data objects based on pair wise similarities, which can be viewed as homogeneous relations (represented by an affinity matrix).

Different other literature algorithms can be viewed as particular relational clustering.

In this thesis we have classified them into two principal categories:

- “*Structure driven approaches*” that are bound to data structure. Data analysis problem is tackled from several dimensions: clustering concurrently columns and rows of a given dataset, like biclustering algorithm presented in subsection 3.1 or vertical 3-D clustering presented in subsection 3.2. These kinds of clustering algorithms can be formulated as clustering on bi-type relational data consisting of only homogeneous relations.
- “*Knowledge driven approaches*” where domain information is used to drive the clustering process and interpret its results: semi-supervised clustering (presented in subsection 4), which is a special type of clustering using both labelled and unlabeled data, has attracted significant attention and is the most significant approach in this category. This kind of clustering algorithms represents the first step to implement a general framework taking into consideration heterogeneous relations and so a real “relational clustering” algorithm.

Consequently, relational data present not only huge challenges to traditional unsupervised clustering approaches, but also great need for theoretical unification of various clustering tasks.

The thesis work focuses on developing a unified framework for relational data clustering and effective algorithms for different types of data from a wide range of applications. The proposed relational approach will be presented in section 1.

### 3. “Structure Driven” Methods

In this section we describe the two main structure driven approaches: biclustering algorithms and three dimensional algorithms.

Both approaches work only based on dataset structure without an explicit integration of unstructured domain knowledge. Biclustering, like we will highlight in subsection 3.1, is a data mining technique that allows a simultaneous clustering of columns and rows of a data matrix.

On the other side, three dimensional clustering, a more recent approach described in section 3.2, aims to concurrently cluster two datasets that share a common set of row labels, but whose column labels are distinct. The resulting clusters reveal the underlying connections between the elements of all three sets of labels.

Like we have already mentioned before, these kinds of clustering algorithms can be viewed as a particular case of “relational clustering”.

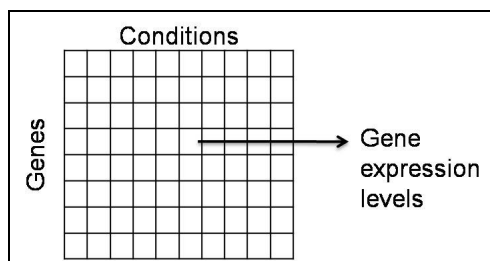
#### 3.1 *Biclustering Algorithm*

In this subsection is proposed a brief survey of existing biclustering algorithms applied to life science domain.

Generally, biclustering is a data mining technique which allows simultaneous clustering of rows and columns of a data matrix. The term became popular thanks to a work by (Hartigan, 1972), although the technique was originally introduced by (Mirkin, 1996). More recently biclustering has been successfully applied by (Cheng and Church, 2000) in the field of gene expression analysis.

The simple idea of a biclustering algorithm is: given an  $m \times n$  matrix, the algorithm generates biclusters, i.e. a subset of rows which exhibit similar behaviour across a subset of columns, or vice versa.

One of the most important applications of clustering algorithm in life science domain is related to the analysis of gene expression dataset. A gene expression dataset contains measurements of increasing or decreasing expression levels of a set of genes. A number of gene expression measurements are usually taken, across time points, tissue samples, or patients. A gene expression dataset (visible in Fig. 3-1) is represented as a matrix of numerical values: gene expression versus experimental condition, gene expression versus tissue, gene expression versus patient.



**Fig. 3-1:** gene expression dataset

The most common objectives pursued during gene expression data analysis, include:

- 1) Grouping of genes according to their expression under multiple conditions.
- 2) Classification of a new gene, given its expression and the expression of other genes, with known classification.
- 3) Grouping of conditions based on the expression of a number of genes.
- 4) Classification of a new sample, given the expression of the genes under that experimental condition.

Traditional clustering techniques can only be used to group either genes or conditions, and, therefore, to pursue directly objectives (1) and (3), above, and, indirectly, objectives (2) and (4).

However, many activation patterns are common to a group of genes only under specific experimental conditions. In fact, our general understanding of cellular processes leads us to expect subsets of genes to be co-regulated and co-expressed only under certain experimental conditions while behaving almost independently under other conditions. Discovering such local expression patterns may be the key to uncovering many genetic pathways that are not apparent otherwise. It is therefore highly desirable to develop new algorithmic approaches capable of discovering local patterns in microarray data (Ben-Dor et al., 2002) and in other kind of data.

These new approaches take the name of biclustering methods that, like just said above, perform clustering in the two dimensions simultaneously.

One of the main differences between clustering and biclustering approaches is that clustering methods derive a *global model*, while biclustering algorithms produce a *local model*. Indeed, each gene in a bicluster is selected using only a condition of the features and each condition in a bicluster is selected using only a subset of the genes.

The resulting clusters do not need to be exclusive and/or exhaustive: a gene or a condition should be able to belong to more than one cluster or to no cluster at all.

### **Definition and problem formulation**

In the case of a gene expression matrix  $A$  whose elements  $x_{ij}$  represents the expression level of gene  $i$  under condition  $j$ , where  $i=1..n$  and  $j=1..m$ .

Such a matrix  $A$ , with  $n$  rows and  $m$  columns, is defined by its set of rows,  $R=\{r_1, \dots, r_n\}$ , and its set of columns,  $C=\{c_1, \dots, c_m\}$ . We will use  $(R,C)$  to denote the matrix  $A$ .

If  $I \subseteq R$  and  $J \subseteq C$  are subsets of the rows and columns, respectively,  $A_{IJ} = (I,J)$  denotes the sub-matrix  $A_{IJ}$  of  $A$  that contains only the elements  $x_{ij}$  belonging to the sub-matrix with set of rows  $I$  and set of columns  $J$ .

Given the data matrix  $A$ , a cluster of rows is a subset of rows that exhibit similar behaviour across the set of all columns. This means that a row cluster  $A_{IC} = (I,C)$  is a subset of rows defined over the set of all columns  $C$ , where  $I = \{i_1, \dots, i_k\}$  is a subset of rows ( $I \subseteq R$  and  $k \leq n$ ). A cluster of rows  $(I; C)$  can thus be defined as a  $k$  by  $m$  sub-matrix of the data matrix  $A$ .

Similarly a cluster of columns is a subset of columns that exhibit similar behaviour across the set of all rows. In this case a cluster is a subset of columns defined over the set of all rows  $R$ , where  $J = \{j_1, \dots, j_s\}$  is a subset of columns ( $J \subseteq C$  and  $s \leq m$ ). A cluster of columns  $(R,J)$  can then be defined as an  $n$  by  $s$  sub-matrix of the data matrix  $A_{RJ}$ .

A bicluster is a subset of rows that exhibit similar behaviour across a subset of columns, and vice-versa. A bicluster  $(I, J)$  results therefore in a  $k$  by  $s$  sub-matrix of the data matrix  $A$ .

The specific problem addressed by biclustering algorithms can now be defined. Given a data matrix,  $A$ , we want to identify a set of biclusters  $B_k = (I_k, J_k)$  such that each bicluster  $B_k$  satisfies some specific characteristics of homogeneity.

The exact characteristics of homogeneity that a bicluster must obey vary.

There are four major classes of biclusters:

- Biclusters with constant values.
- Biclusters with constant values on rows or columns.
- Biclusters with coherent values.
- Biclusters with coherent evolutions.

Below a description of each class is reported.

***Biclusters with Constant Values***

This approach only produces good results when it is performed on ordered data (rows and columns must be reordered) and on non-noisy data. A *perfect* constant bicluster (visible in Fig. 3-2) is a sub-matrix  $(I, J)$ , where all values within the bicluster are equal for all  $i \in I$  and all  $j \in J$ :

$$A_{ij} = \mu \tag{3-1}$$

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

**Fig. 3-2:** Perfect constant bicluster

Different approaches are present in literature like that proposed in (Hartigan, 1972) which introduced a partition based algorithm called *Block Clustering*.

***Biclusters with Constant Values on Rows or Columns***

The biclusters in Fig. 3-3(a) and Fig. 3-3(b) are examples of biclusters with constant rows and constant columns, respectively.

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

(a) Bicluster with constant rows

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

(b) Bicluster with constant columns

**Fig. 3-3:** Bicluster with constant values on rows or columns

A *perfect* bicluster with constant rows or columns is a sub-matrix where all the values within the bicluster can be obtained using one of the following additive or multiplicative models:

$$\begin{aligned} x_{ij} &= \mu + \alpha_i & \text{or} & & x_{ij} &= \mu + \beta_j \\ x_{ij} &= \mu \times \alpha_i & & & x_{ij} &= \mu \times \beta_j \end{aligned} \tag{3-2}$$

where  $\mu$  is the typical value within the bicluster and  $\alpha_i$  is the adjustment for row  $i \in I$  and  $\beta_j$  is the adjustment for column  $j \in J$ . These adjustments can be obtained either in an additive or multiplicative way.



Many biclustering algorithms aim at finding these types of biclusters: (Getz et al., 2000) introduced the *Coupled Two-Way Clustering (CTWC)* algorithm, (Sheng et al., 2003) tackled the biclustering problem in a Bayesian framework.

**Biclusters with Coherent Values**

When an additive or multiplicative model is used within the biclustering framework, a *perfect* bicluster with coherent values,  $(I, J)$ , is defined as a subset of rows and a subset of columns, whose values  $x_{ij}$  are predicted using the following expression:

$$x_{ij} = \mu + \alpha_i + \beta_j \text{ or } x_{ij} = \mu \times \alpha_i \times \beta_j \tag{3-3}$$

where  $\mu$  is the typical value within the bicluster,  $\alpha_i$  is the adjustment for row  $i \in I$  and  $\beta_j$  is the adjustment for column  $j \in J$ .

The bicluster in Fig. 3-4(a) is an example of a bicluster with coherent values on both rows and columns, whose values can be described using an additive model.

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

(a) Additive model

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

(b) Multiplicative model

**Fig. 3-4:** Bicluster with coherent values

Several biclustering algorithms attempt to discover biclusters with coherent values assuming either additive or multiplicative models: (Cheng and Church, 2000) introduced a *mean squared residue* as a measure of the coherence of the rows and columns in the bicluster. (Klugar et al., 2003) looked for checkerboard structures in the data matrix by integrating biclustering of rows and columns with normalization of the data matrix. (Tang et al., 2001) introduced the *Interrelated Two-Way Clustering (ITWC)* algorithm that combines the results of one-way clustering on both dimensions of the data matrix in order to produce biclusters. (Lazzeroni and Owen, 2000) introduce the plaid model where the value of an element in the data matrix is viewed as a sum of terms called *layers*.

**Biclusters with Coherent Evolutions**

These biclustering algorithms address the problem of finding coherent evolutions across the rows and/or columns of the data matrix regardless of their exact values. The co-evolution property can be observed on both rows and columns of the biclusters, as it is shown in Fig. 3-5 (a), on the rows of the bicluster or on its columns. The biclusters presented in Fig. 3-5 (c) and Fig. 3-5(d) are examples of biclusters with coherent evolutions on the columns, while Fig. 3-5 (b) shows a bicluster with co-evolution on the rows.

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

(a) Overall coherent evolution

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

(b) Coherent evolution on the rows

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

(c) Coherent evolution on the columns

70	13	19	10
29	40	49	35
40	20	27	15
90	15	20	12

(d) Example of bicluster with coherent evolution on the columns

**Fig. 3-5:** Bicluster with coherent evolution

(Ben-Dor et al., 2002) defined a bicluster as an order-preserving sub-matrix (OPSM), a group of rows whose values induce a linear order across a subset of the columns. (Murali and Kasif, 2003) assumed that data may contain several *xMOTIFs* (biclusters) and aimed at finding the largest *xMOTIF*: the bicluster that contains the maximum number of conserved rows.

In conclusion, the simplest biclustering algorithms identify subsets of rows and subsets of columns with constant values. More sophisticated approaches look for biclusters with coherent values on both rows and columns: each row and column can be obtained by adding a constant to each of the others or by multiplying each of the others by a constant value.

According to the specific properties of each problem, one or more of these different types of biclusters are generally considered interesting.

### 3.2 3- Clustering Algorithm

3-Clustering goes one step beyond biclustering and aims to concurrently cluster two datasets that share a common set of row labels, but whose column labels are distinct. Such clusters reveal the underlying connections between the elements of all three sets of labels.

To outline the main advantages of the 3D approach we consider a “toy” example with references to the NCI-60 dataset (see Appendix A), composed by two matrices  $T$  and  $A$  containing cell lines and gene expression and cell lines and drug activity data, respectively. Just for the sake of this example we have assumed a cut-off value for gene expression and drug activity levels, so that the matrices are binary, like represented in Fig. 3-6 (a) and (b).

Therefore, in  $T$  element  $t_{ij}$  is equal to 1 if gene  $j$  has a gene expression values over the fixed threshold, 0 otherwise. The same thing is done for matrix  $A$  where the value of  $a_{ij}$  cell is equal to 1 if the response of drug  $j$  is greater than a prefixed threshold and 0 otherwise.

	$G_1$	$G_2$	$G_3$	$G_4$
$C_1$	1	1	1	0
$C_2$	1	0	0	0
$C_3$	1	1	0	0
$C_4$	1	1	0	1

	$D_1$	$D_2$	$D_3$	$D_4$
$C_1$	1	0	1	1
$C_2$	0	1	1	0
$C_3$	1	0	0	0
$C_4$	0	1	0	0

(a)  $T$ : Cell lines ( $C_1... C_4$ ) vs. Genes ( $G_1... G_4$ )

(b)  $A$ : Cell lines ( $C_1... C_4$ ) vs. Drug response ( $D_1... D_4$ )

$\langle \{G_1\}, \{C_1, C_2, C_3, C_4\} \rangle$
$\langle \{G_1, G_2\}, \{C_1, C_3, C_4\} \rangle$
$\langle \{G_1, D_1, G_2\}, \{C_1, C_3\} \rangle$
$\langle \{G_1, D_3\}, \{C_1, C_2\} \rangle$
$\langle \{G_1, D_2\}, \{C_2, C_4\} \rangle$
$\langle \{G_1, D_1, G_2, G_3, D_3, D_4\}, \{C_1\} \rangle$
$\langle \{G_1, G_2, G_4, D_2\}, \{C_4\} \rangle$

(c) Clusters in the join of  $T$  and  $A$

$\langle \{G_1, G_2, G_3\}, \{D_1, D_3, D_4\}, \{C_1\} \rangle$
$\langle \{G_1, G_2, G_4\}, \{D_1\}, \{C_4\} \rangle$

(d) 3-clusters across  $T$  and  $A$

Fig. 3-6: Sample data and clusters

In principle we could join  $A$  e  $T$  in a single matrix  $D = [A|T]$ . Fig. 3-6 (d) shows the 3-clusters that connect both  $A$  and  $T$  matrices, while figure Fig. 3-6 (c) shows clusters for  $D$ .

For example consider the clusters listed in Fig. 3-6(c). The first two clusters  $\langle \{G_1\}, \{C_1, C_2, C_3, C_4\} \rangle$  and  $\langle \{G_1, G_2\}, \{C_1, C_3, C_4\} \rangle$  correspond to biclusters found within  $T$  and thus do not reveal any association between drug responses and gene expressions. The next three clusters  $\langle \{G_1, D_1, G_2\}, \{C_1, C_3\} \rangle$ ,  $\langle \{G_1, D_3\}, \{C_1, C_2\} \rangle$  and  $\langle \{G_1, D_2\}, \{C_2, C_4\} \rangle$  do reveal some associations between drugs and gene expressions.

However, if we partition each cluster, the partitions do not correspond to local bi-clusters in the individual datasets. In other words consider  $\langle \{G_1, D_3\}, \{C_1,$

$C_2\}$ >, the partition  $\langle\{G_1\}, \{C_1, C_2\}\rangle$  is not a bi-cluster in  $T$  (since it is enclosed within the larger pattern  $\langle\{G_1\}, \{C_1, C_2, C_3, C_4\}\rangle$ ).

Still in this small "toy" example we see that only two of the seven mined bi-clusters in the join of  $T$  and  $A$  revealed associations between the clusters of  $T$  and  $A$ .

If we consider instead the 3-cluster  $\langle\{G_1, G_2, G_3\}, \{D_1, D_3, D_4\}, \{C_1\}\rangle$ , it not only points out an association among genes  $G_1, G_2, G_3$  and drug responses  $D_1, D_3, D_4$  through the cell line  $C_1$ , but also an association among the cell line  $C_1$  and genes  $G_1, G_2, G_3$  through drug responses terms  $D_1, D_3, D_4$ .

Thus 3-clusters not only disclose associations between attributes of individual datasets through objects, but also reveal associations between local biclusters with respect to attributes in other data-sets.

For a formal presentation of 3-Clustering see (Alqadah and Bhatnagar, 2008).

### 3.3 Quality Measures for Biclustering and 3-Clustering

In this subsection we report quality measures for the structure driven methods that we have just presented. In particular, the proposed measures for both approaches have been defined only for binary matrices (Procopiu et al., 2002; Alqadah and Bhatnagar, 2008).

#### 3.3.1 Quality Measure for Biclustering

As already defined above, biclusters may be thought as rectangular sub-matrices of the original dataset, in which the number of objects correspond directly to the height of the rectangle, while the number of items corresponds directly to its width.

Given a bicluster  $A_{IJ}=(I, J)$  where  $I$  denote the height of  $A_{IJ}$  and  $J$  denote the width of  $A_{IJ}$ . The number of 1's in  $A_{IJ}$  then corresponds directly to the area enclosed by  $A_{IJ}$ , which we denote as:

$$\rho(C) = |I| * |J| \tag{3-4}$$

Utilizing  $\rho$  as a quality measure for local bi-clusters has two major drawbacks:

- a) it does not distinguish between the individual contribution of width and height to the total number of 1's in a pattern.
- b)  $\rho$  does not take into account the fact that as height increases, the width of a bicluster must decrease.

In order to overcome these drawbacks, (Alqadah and Bhatnagar, 2008) introduce a parameter  $\beta$  which represents a trade-off value for the percentage of items we are willing to drop for each additional object added. Equivalently  $\beta$  represents how many units of width we are willing to drop for each additional unit of height.

It's now possible to construct a quality measure,  $\Omega$  centred around  $\beta$  and  $\rho(A_{IJ})$ . Formally, given a bi-cluster  $A_{IJ}$  in a dataset  $D_i$  and  $\beta$  ( $0 < \beta < 1$ ), let:

$$\Omega_i(A_{IJ}) = \mu_i(|I|, |J|) \tag{3-5}$$

denotes the quality of  $A_{IJ}$  in  $D_i$ . Then  $\Omega_i$  should:

$$\begin{aligned} & \text{maximize } \rho(A_{ij}) \\ & \text{subject to } \mu_i(|I|, |J|) = \mu_i(\beta^* |I|, |J| + 1) \end{aligned} \quad (3-6)$$

In order to solve the optimization problem defined above function  $\mu(a, b)$  must satisfy the following two conditions (Procopiuc et al., 2002):

1.  $\mu(a, b)$  should be monotonically increasing in both  $a$  and  $b$
2.  $\mu(a, b)$  should be a  $\beta$ -balanced. Mathematically this implies that:

$$\mu(a, b) = \mu(\beta^n a, b + n), 0 < \beta < 1, n \in N^+ \quad (3-7)$$

One such function that satisfies conditions 1 and 2 is:

$$\mu(a, b) = a \left(\frac{1}{\beta}\right)^b \quad (3-8)$$

This results in the following definition for the quality of a bicluster. Given  $\beta$  and a bi-cluster  $A_{ij}=(I, J)$  in the dataset  $D_i$  its quality  $\Omega_i(A_{ij})$  is given by

$$\Omega_i(A_{ij}) = \mu_i(w(C), h(C)) = |X| \left(\frac{1}{\beta}\right)^{|Y|} \quad (3-9)$$

### 3.3.2 Quality Measure for 3-Clustering

Usually when 3-Clustering algorithm is applied to binary matrices, we would like to:

- ✓ Maximize the number of one's in a 3-cluster.
- ✓ Maximize the number of objects and items.

A quality measure for 3-clusters has been developed in (Alqadah and Bhatnagar, 2008) using the same intuition that was used for biclusters. 3-Clusters may also be thought of as rectangular sub-matrices across two data-tables.

Therefore, given a 3-cluster  $C_{12} = \langle X, Y, Z \rangle$ , where  $X$  is a subset set of the feature of data matrix  $D_1$ ,  $Y$  represents the features' subset of the other data matrix  $D_2$  and  $Z$  is a subset of the instances common to both data matrices:

$$\rho(C_{12}) = (|X| + |Y|) * |Z| \quad (3-10)$$

As the height of a 3-cluster increase, its width also must decrease, just as was the case with biclusters. Utilizing this fact, and properties 1 and 2 from the previous subsection we may now derive  $\Omega_{12}$ :

$$\begin{aligned} \Omega_{12}(C^{12}) &= \mu_{12}(|X| + |Y|, |Z|) \\ &= (|X| + |Y|) * \left(\frac{1}{\beta}\right)^{|Z|} \\ &= |X| \left(\frac{1}{\beta}\right)^{|Z|} + |Y| \left(\frac{1}{\beta}\right)^{|Z|} \\ &= \Omega_1(\langle X, Z \rangle) + \Omega_2(\langle Y, Z \rangle) \end{aligned} \quad (3-11)$$

The above equation is clearly  $\beta$ -balanced, and can be computed since it is the sum of the quality of bi-clusters.

In this way, given  $\beta$  and a 3-cluster  $C_{12} = \langle X, Y, Z \rangle$  across  $D_i$  and  $D_j$  its quality  $\Omega_{ij}(C)$  is given by

$$\Omega_{ij}(C) = \Omega_i(\langle X, Z \rangle) + \Omega_j(\langle X, Z \rangle) \quad (3-12)$$

The value of  $\beta$  will have a great effect on the nature of 3-clusters discovered. Higher values of  $\beta$  will favour 3-clusters containing more items from either  $D_1$  or  $D_2$  and fewer objects. As mentioned earlier,  $\beta$  represents the trade-off between how many columns a user is willing to give up in order to include  $n$  more rows.

## 4. “Knowledge Driven” Methods

A large quantity of *unlabeled data* is available in many real-life data mining tasks, e.g., genes of unknown functions, uncategorized messages in an automatic email classification system, etc; on the contrary, *labelled data* is often limited and expensive to generate, since labelling typically requires human expertise.

The first fact explains why clustering is common as an exploratory data analysis, the second why *semi-supervised learning* has become a topic of significant recent interest (Blum and Mitchell, 1998; Joachims, 1999; Nigam et al., 2000).

In this section, we outline the main results on semi-supervised clustering, where the performances of unsupervised clustering algorithms are improved with limited amounts of supervision in the form of labels on the data or constraints (Wagstaff et al., 2001; Basu et al., 2003a; Klein et al., 2002; Xing et al., 2003; Basu et al., 2003b). In particular, *semi-supervised learning* can be viewed as a special case of relational clustering in which instances are represented into a propositional form and relationships among them are retained.

Generally a fully unsupervised clustering algorithm might naturally find a solution that is consistent with the domain knowledge; the most interesting cases are those in which the domain knowledge suggests that the default solution is not the one that is sought. Therefore, researchers began exploring principled methods of enforcing desirable clustering properties.

Recently semi-supervised clustering algorithms have been proposed that can incorporate pair wise constraints on cluster membership (Demiriz et al., 1999; Wagstaff et al., 2001; Basu et al., 2002) or learn problem-specific distance metrics that produce desirable clustering output (Cohn et al., 2003; Bilenko and Mooney, 2003; Hertz et al., 2004; Chang and Yeung, 2004; Bar-Hillel et al., 2005). This research area has been expanded to include algorithms that leverage many additional kinds of domain knowledge for the purpose of clustering (Basu et al., 2004).

Therefore, existing methods for semi-supervised clustering can be classified into three general categories usually called *constraint-based*, *distance-based* and *hybrid* clustering. The last one aims at defining a framework able to combine distance and constraint based methods.

In the next subsections we provide a current account of the innovations in these three semi-supervised clustering categories.

### 4.1 Constraint - Based Clustering Methods

In *Constraint-based* clustering problems some pre-existing knowledge about the desired partitioning is available. This knowledge can be provided by the user in the form of labels or constraints to guide the clustering algorithm towards a more appropriate data partitioning.

Constrained clustering was first introduced by using instance-level constraints. In particular, a set of instance-level constraints,  $C$ , consists of statements about pairs of instances (or objects). If two instances should be placed into the same cluster, a *must-link* constraint between them is expressed as  $c = (i,j)$ . Likewise, if two instances should not be placed in the same cluster,  $c \neq (i,j)$ , express a *cannot-*

*link* constraint. If constraints are available, rather than returning partitions that satisfy the generic objective function used by the clustering algorithm, it is required that the algorithm adapts its solution to accommodate  $C$ .

These instance-level constraints have several interesting properties.

A collection of must-link constraints encodes an equivalence relation (symmetric, reflexive and transitive) on the instances involved. The transitivity property permits additional must-link constraints to be inferred from the base set. More generally, if we produce a graph in which nodes represent instances and edges represent must-link relationships, then any must-link constraint that joins two connected components will entail an additional must-link constraint between all pairs of items on those components. In contrast, the cannot-link constraints do not encode transitivity; indeed  $c = (i,j)$  and  $c \neq (j,k)$  implies  $c \neq (i,k)$ .

The full set of constraints can be used in a variety of ways, including enforcing individual constraints and using them to learn a problem-specific distance metric.

So we can say that *constraint-based* methods rely on user-provided labels or constraints to guide the algorithm towards a more appropriate data partitioning. In literature different approaches apply these methods in different ways: by modifying the objective function for evaluating clustering so that it includes satisfying constraints, like in (Demiriz et al., 1999); or by enforcing constraints during the clustering process like in the COP-KMeans algorithm proposed in (Wagstaff et al., 2001), or initializing and constraining the clustering based on labelled examples (Basu et al., 2004a).

## 4.2 Distance - Based Clustering Methods

*Distance-based* approaches are characterized by clustering distortion measures which take into account the labels or constraints in the supervised data.

These can be viewed as other fruitful approaches to incorporate constraints as statements to adjust the distance (or similarity) between instances for accommodating the given relationship between instances. Indeed, a must link constraint  $c=(i,j)$  implies that instances  $i$  and  $j$  should be close together and a cannot-link constraint  $c \neq (i,j)$  implies that they should be sufficiently far apart to never be clustered together.

Research on distance-based semi-supervised clustering with pairwise constraints includes the work of (Cohn et al., 2003), who used gradient descent for weighted Jensen-Shannon divergence in the context of Expectation Maximization (EM) clustering; (Xing et al., 2003) utilized convex optimization and iterative projections to learn a Mahalanobis distance for K-means clustering; the Redundant Component Analysis (RCA) algorithm used only must-link constraints to learn Mahalanobis distance using convex optimization (Bar-Hillel et al., 2005). Other methods include training a string-edit distance using EM (Bilenko and Mooney, 2003), modification of the squared Euclidean distance using the shortest path algorithm (Klein et al., 2002), learning a margin-based clustering distortion measure using boosting (Hertz et al., 2004), and learning a distance metric transformation that is globally linear but locally non-linear (Chang and Yeung, 2004).



Most of these distance learning techniques for clustering learn the distance measure first using only supervised data, and then perform clustering on the unsupervised data.

### 4.3 Hybrid Methods: Combination of Distance-Based and Constraint-Based

As we have described in the last subsections, existing methods for semi-supervised clustering can be generally grouped into *constraint-based* methods, with the aim to guide the clustering process with pairwise instances, and *distance-based* methods, that employ metric learning techniques to get an adaptive distance measure to be used in the clustering process.

A *hybrid method*, that combines these two methods under a single probabilistic framework, has been proposed by (Basu et al., 2004). In particular, authors present a general framework based on *Hidden Markov Random Fields* (HMRFs) that combines the constraint-based and distance-based approaches in a unified model.

This semi-supervised clustering model considers a set of data points  $X$  with a specified distance measure  $D$  between the points. Supervision is provided as a set of must-link constraints  $c=(i,j)$  (with a set of associated violation costs  $W$ ) and a set of cannot-link constraints  $c\neq(i,j)$  (with associated violation costs  $W$ ). The objective is to partition the data into  $K$  clusters so that the total distance between the points and the corresponding cluster centroids according to the given measure  $D$  is minimized while a minimum number of constraints are violated.

An *HMRF* is defined by the following components:

- A *hidden field*  $L = \{l_i\}_{i=1}^N$  of random variables, whose values are unobservable. In the clustering framework, the set of hidden variables are the unobserved cluster labels on the points, indicating cluster assignments. Every hidden variable  $l_i$  takes values from the set  $\{1, \dots, K\}$ , which are the indices of the clusters.
- An *observable set*  $X = \{x_i\}_{i=1}^N$  of random variables, where every random variable  $x_i$  is generated from a conditional probability distribution  $Pr(x_i|l_i)$  determined by the corresponding hidden variable  $l_i$ . The random variables  $X$  are conditionally independent given the hidden variables  $L$ , i.e.,

$$\Pr(X | L) = \prod_{x_i \in X} \Pr(x_i | l_i)$$

In the framework, the set of observable variables for the HMRF corresponds to the given instances.

Relationships between pairs of instances are provided by user supervision and summarized by a relation matrix  $R$  as follows:

$$r_{ij} = \begin{cases} w_{ij}^M d(x_i, x_j) & \text{if } (x_i, x_j) \in M \\ w_{ij}^C d(x_i, x_j) & \text{if } (x_i, x_j) \in C \\ 0 & \text{otherwise} \end{cases} \quad (4-1)$$

Where:

- $w_{ij}^M d(x_i, x_j)$  is a function that penalizes the violation of must-link constraints.
- $w_{ij}^C d(x_i, x_j)$  is a penalty function for cannot-link.

Each relationship contributes to the clustering process, according to the weight of must and cannot constraints violation ( $w_{ij}^M$  and  $w_{ij}^C$ ) smoothed by a penalty scaling function  $d(x_i, x_j)$  defined over the given feature space. In this case must and cannot-links are provided in order to guide the clustering process according to the existing relationships. Indeed, two objects  $x_i$  and  $x_j$  may share either a *must* or a *cannot relationship* if there exists a user supervision which states that  $x_i$  and  $x_j$  should or should not assigned to the same cluster.

Given must and cannot relationships, the clustering objective function is to minimize the objective function as follows:

$$\min \sum_{x_i} d(x_i, c_g) + \sum_{(x_i, x_j) \in M} w_{ij}^M d(x_i, x_j) I_f[l_i \neq l_j] + \sum_{(x_i, x_j) \in C} w_{ij}^C d(x_i, x_j) I_f[l_i = l_j] \quad (4-2)$$

where  $I_f$  is an indicator function that denotes a violation of must or cannot constraints. In this case, if a supervised relationship provided by the user is not respected during the clustering process, the objective function is penalized and the cluster assignments are accordingly refined. A refined version of this model, focused on two-type relational data, has been investigated in the next section, where is described the proposed relational clustering framework.

#### 4.4 The Proposed Relational Clustering Framework:

##### *Principal Features*

In the last subsection, we have outlined different clustering methods aimed at including some information in the clustering process introducing penalization components, which are defined at the beginning of the clustering phase, in the clustering objective function .

Instead, in our proposed relational clustering algorithm, relationships between data are not known a priori but are learned and subsequently used to smooth the assignment process through the penalization of those placements that increment distances between instances. The proposed approach can therefore be classified as a *relational hybrid method*.

The main goal of the algorithm is to find the optimal partitioning of a set of related instances into exclusive clusters through the optimization of an objective

function based not only on features similarity/dissimilarity, but also on the inclusion of information coming from the relationships among instances. These relationships, are not given as an input, but they are learned from background knowledge about instances themselves.

We started from the general clustering framework described in section 2.3 (Fig. 2-10) and modified it to obtain the one represented in Fig. 4-1.

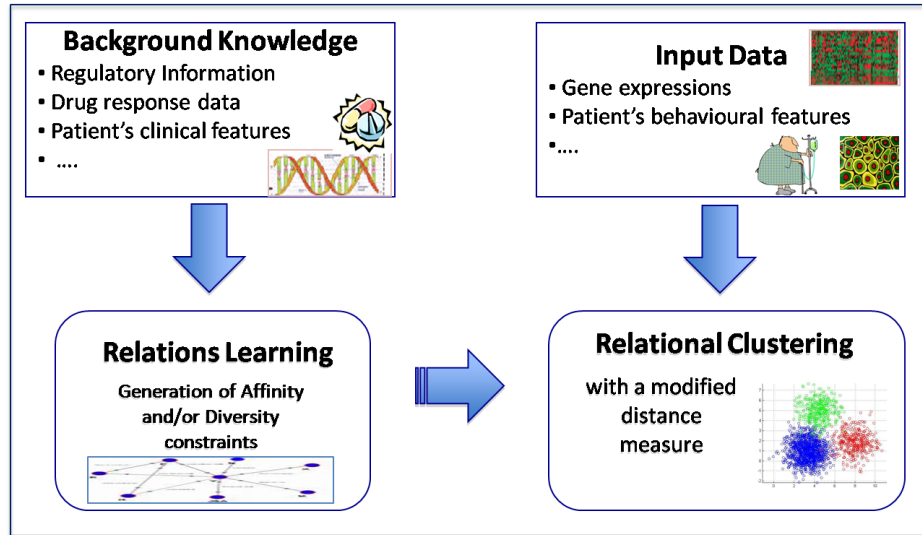


Fig. 4-1: Relational Clustering Core

The proposed clustering procedure is based on two principal phases that take into account both input and background data for relationships learning. These phases are:

1. *Relations Learning*: in general in this phase relationships are discovered and learned from domain background information.
2. *Relational Clustering*: in this phase, the learned relationships are included into the clustering process modifying the clustering objective function.

The relation learning phase receives as input background information about instances in the form of a data matrix, where each instance is characterized by a set of features. Generally, in the proposed framework relations are learned through a traditional clustering algorithm used as an exploratory technique to better analyze additional information and find possible relationships between instances. So in this relation learning step we can define two principal relationship types among instances, each one represented by a different degree of intensity. We will refer to these as *Affinity* and *Diversity* link (with the word link we mean relationships).

In particular, an *Affinity link* is a link between two instances belonging to the same cluster in this first step. This link suggests us that maybe there is a relation between these instances, since they share some similar features and this relationship has a weight equal to the distance between these. In this way, if two instances are more similar each other (they are very close in the features space) they are likely to be related.

In the opposite way, a *Diversity link* is a link between two instances belonging to different clusters in this first step. As a consequence, this kind of links suggests

us that probably there isn't a relationship between these two instances or the instances have a very weak relationship.

A simple representation of Affinity and Diversity Links is illustrated in Fig. 4-2. In the left part of the figure a simple data representation of background information is provided, where each instance is represented by a vector of  $n$  features (as already presented in section 1 and 3).

The affinity and diversity links are visible on the right of the figure, where a particular weight based on distances between instances is assigned to each link. For example, the link among instance  $i$  and  $j$  is an affinity link, and links between instances  $z$  and  $h$  or instances  $g$  and  $f$  are diversity links.

It is necessary to point out that the two diversity links have different weights. In this way, we can so define two matrices of relations  $R^A$  (for affinity relations) and  $R^D$  (for diversity relations) where each element of the matrices represents the links weight. Formally:

$$r_{ij}^A = \begin{cases} d_{ij} & \text{if instance } i \text{ and instance } j \text{ belong to the same cluster } C \\ 0 & \text{otherwise} \end{cases} \quad (4-3)$$

$$r_{ij}^D = \begin{cases} d_{ij} & \text{if instance } i \in \text{cluster } C_1 \text{ and instance } j \in \text{cluster } C_2 \\ 0 & \text{otherwise} \end{cases} \quad (4-4)$$

From the above definition we can see that in our approach the weight of a relationship between two instances depends on the distance between them. This distance will be computed with one of the different measures presented in subsection 1.1 chosen according to the application domain.

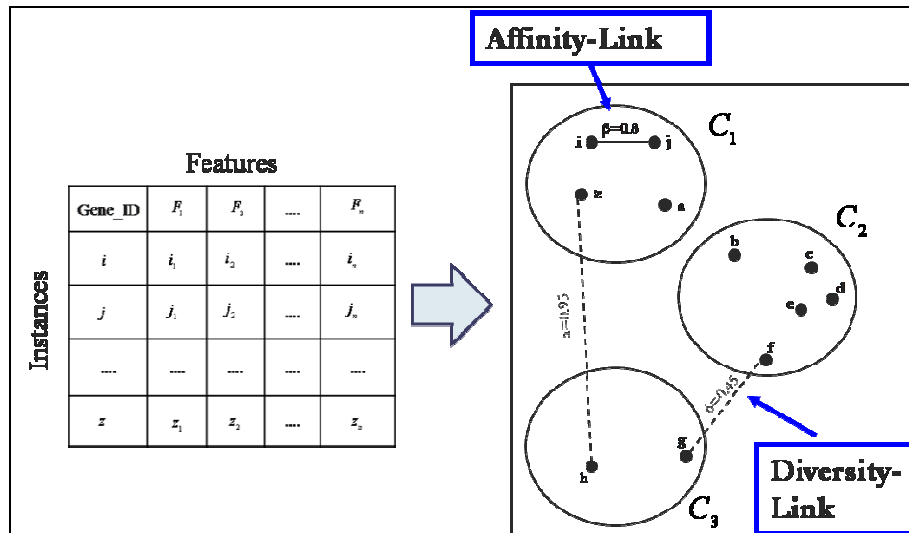


Fig. 4-2: Relation Learning Phase

In this way, the strength of the relationships will be used to modify, in the *relational clustering phase*, the traditional clustering objective function presented in subsection 1.2.

Defining the assignment matrix  $Z$  (as in ( 1-10)):

$$z_{ij} = \begin{cases} 1 & \text{if instance } i \in C_n \\ 0 & \text{otherwise} \end{cases} \quad (4-5)$$

The clustering problem can be formulated as:

$$\min \sum_{c=1}^C \left[ \sum_{i,j} \left[ d_{ij} z_{ic} z_{jc} - d_{ij} z_{ic} (1 - z_{jc}) + z_{ic} z_{jc} r_{ij}^D d_{ij} + z_{ic} (1 - z_{jc}) r_{ij}^A d_{ij} \right] \right] \quad (4-6)$$

This optimization problem can be solved through an adaptation of K-means heuristic, reported in Algorithm 1-1.

As we can understand from equation ( 4-6), if an affinity or a diversity link between instances  $i$  and  $j$  is not preserved, the objective function is penalized according to the weight defined  $r_{ij}^A$  (or  $r_{ij}^D$ ).

In this subsection we have provided only a general description of the proposed relational clustering framework.

In this thesis this framework has been developed and applied for different kind of applications in life science domain. As a consequence, this general relational framework will be modified and instantiated for each application, based on the goal of the application itself.

In the next chapter three different instantiations of the framework are presented.

## 5. The Proposed Relational Clustering Framework:

### Case Studies

This chapter describes the proposed relational clustering framework applied to three different case studies:

- Learning transcriptional regulatory modules (section 5.1)
- Detecting the most effective cancer drug (section 5.2)
- Oral Anticoagulation Therapy (OAT) (section 5.3)

#### 5.1 Learning Transcriptional Regulatory Modules

In this section we present a particular instantiation of the relational clustering framework for finding modules of co-regulated genes, i.e. group of genes that are likely to be regulated by the same set of transcription factors.

The main results of this analysis are synthesized in (Archetti et al., 2008; Archetti et al., 2009c).

##### 5.1.1 The biological problem: state of the art

To completely understand the behaviour of an organism, an organ, or still a single cell, we need to learn the underlying regulatory mechanisms governing the expression of genes in the cell itself. A key junction in these mechanisms is the mRNA transcription regulated by proteins, known as *transcription factors* (TF). A TF binds to specific DNA sequences in the promoter region of a target gene and activates the transcription process from DNA to RNA. A consequence of this particular mechanism is the expression of the target gene in a cell. Genes regulated by a common set of TFs are usually called co-regulated genes, since their expression levels will follow the same pattern.

Understanding the regulatory mechanisms of gene transcription is an important goal of molecular biology and high-throughput genomics and the availability of DNA microarrays, which measure the expression level of thousands of genes simultaneously, has given an important contribution in this direction (Haverty et al., 2004).

Several approaches have been proposed to uncover genes' regulatory mechanisms, by partitioning genes into *transcriptional modules* (TM), i.e. groups of genes that obey to a common transcriptional program.

The most commonly used computational methodology for the discovery of co-regulated genes applies clustering algorithms to expression data and then searches for the transcription factors that most probably bind to each set of co-expressed genes (Eisen et al., 1998; Liu et al. 2001; Sinha et al. 2000). Biclustering algorithms, already presented in section 3.1, have also been applied (Bergmann et al. 2003; Ihmels et al. 2004), aimed at obtaining biclusters of co-regulated genes, i.e. clusters characterized by a set of genes and the set of experimental conditions that induce their co-regulated expression, thus facilitating the identification of set of genes involved in the same functional role.

In this case however the identification of the transcription factors involved in the regulation is not directly revealed and is left to a further analysis. (Middendorf et al., 2005) introduced a particular motif discovery algorithm called MEDUSA (Motif Element Discrimination Using Sequence Agglomeration) that builds motifs models whose presence in the promoter region of a gene is predictive of differential expression. In particular, putative binding sites are used to build a decision tree that tries to explain the gene expression profiles in terms of motifs. Other approaches like that by (Barash et al. 2001), starting from the analysis of sequence data, work in the opposite direction: they first reduce the sequence data into some predefined features of the gene, given by the presence or absence of various potential transcription factor binding sites, then identify groups of genes having common patterns.

All these clustering-based methods principally partition genes into mutually exclusive clusters measuring correlative and linear relationship among genes and discover TM by presuming that genes with similar expression profiles share similar functions. However, genes implicated in the same biological process can have different expression patterns (Zhou et al., 2005; Zhang et al., 2004).

Recently, more sophisticated algorithms have been presented to attempt to combine both upstream sequence information and expression data in a single framework to build prediction models able to identify regulation patterns: in particular (Segal et al. 2003) constructed a probabilistic model that uses expression data to link regulators to regulated genes. Their method relies on the assumption that the expression levels of regulated genes will depend on those of regulators, which is a limitation in cases in which the expression level of the regulator does not change appropriately (e.g., cases of post-transcriptional modification). Moreover, they might produce gene clusters that are not biologically interpretable because both the cluster and the regulation program are free parameters that have to be optimized.

(Ernst et al., 2008), propose SEREND, a semi-supervised regulatory network discovery method applied to the bacterium *Escherichia Coli*. They use an iterative classification scheme that exploits a compendium of gene expression data in a semi-supervised way in order to predict novel regulator-target interactions: they train a model using expression and regulatory motif data and then infer novel interactions using their model on expression and motif data.

These methods aim at finding new motifs that are probably involved in the regulation of the uncovered clusters of genes or new predictions of transcription factors-genes interactions.

In the last years, large databases about known transcriptional factors and their functionalities have been made available for an increasing number of organisms. Taking into account this information we can associate to each gene a TF profile, whose elements represent the strength of the relation between genes and TFs.

(Clements et al. 2007), for example, exploit this information to cluster genes using a weighted distance measure to combine features associated to gene expression levels with features associated with information about interaction between gene and known TF.

Analyzing all the clustering-based approaches just presented, we have seen that generally they aim at partitioning genes into a specified number of clusters through the minimization of a cost function related to a similarity/dissimilarity measure between points computed usually on the basis of a distance measure,

without taking into account background and contextual information about pairs of instances for constraining their cluster placement.

By using the available data (like microarray, dna-binding, protein-protein interaction and sequence data) in an integrative framework it is possible to unravel the regulation process at a more global level.

The conclusions drawn by previous investigations and the weakness of traditional distance-based algorithms, lead us to the application of our relational clustering framework taking into account the background knowledge. In this way, the incorporation of this knowledge permits to obtain an improved and reliable picture of the whole transcriptional regulation process.

In particular, in this application we develop an instantiation of the general relational clustering framework (illustrated in section 4.4): we propose an *iterative relational clustering procedure* that given a dataset  $\Omega$  of genes, at each iteration, determines a set of possible significant regulatory interactions by identifying a set of candidate transcription factors and then refines the clusters of genes based on their expression level, by modifying the distance measure accordingly. As a result we obtain clusters of genes that are both *co-expressed and co-regulated* by the same set of transcription factors.

In the next subsections initially we describe the method used for obtaining gene regulatory information and then we integrate this information with gene expression data thru the iterative relational clustering procedure.

### 5.1.2 Integration of Gene Regulatory Information and Gene Expression Data

Gene regulatory information is computed by using *Pscan*, a recently developed tool (Zambelli et al., 2009). Pscan is a software tool that takes as input a set of candidate co-regulated genes and gives as output their binding values with respect to a set of known TF (order by their associated p-value).

For each gene  $i$  we consider a vector of real numbers  $g_i^{TF}$ , that we call *TF profile* of the gene, whose element  $g_{it}^{TF}$  represents the binding value of gene  $i$  with respect to TF  $t$ .

The result is a set  $\Omega^{TF}$  (an example is visible in Fig. 5-1) that can be regarded as a measure of the relationship strength between genes and the set of TFs.

Gene_ID	$TF_1$	$TF_2$	...	$TF_n$
$g_1$	$g_{11}^{TF}$	$g_{12}^{TF}$	...	$g_{1n}^{TF}$
$g_2$	$g_{21}^{TF}$	$g_{22}^{TF}$	...	$g_{2n}^{TF}$
...	...	...	...	...
$g_{12}$	$g_{n1}^{TF}$	$g_{n2}^{TF}$	...	$g_{nm}^{TF}$

Fig. 5-1:  $\Omega^{TF}$  matrix



**Combining gene expression data and gene regulation: a simple traditional clustering approach**

The dataset can be viewed as a set into the space  $R^{n+m}$  defined as:

$$\Omega = \{ g \mid g = (g^{Exp}, g^{TF}), g^{Exp} \in R^n, g^{TF} \in R^m \} \quad (5-1)$$

where for each gene we have both its expression values in  $n$  experimental conditions  $g^{Exp}$  and its TF profile  $g^{TF}$  with respect to a set of  $m$  known TFs.

We can consequently define  $\Omega^G$  and  $\Omega^{TF}$  as the set of the genes represented thru their gene expression profile and their binding values with known TFs respectively:

$$\Omega^{Exp} = \{ g^{Exp} \mid g = (g^{Exp}, g^{TF}), g \in \Omega \} \quad (5-2)$$

$$\Omega^{TF} = \{ g^{TF} \mid g = (g^{Exp}, g^{TF}), g \in \Omega \} \quad (5-3)$$

In this way a gene  $i \in \Omega$  can be represented as a vector  $g_i \in \mathfrak{R}^{n+m}$ .

A first integration of this data in the clustering procedure can be done following the approach proposed by (Clements et al., 2007), also called STVQ by (Graepel, 1968). They suggest integrating the occurrence of known regulating elements in the upstream region of genes together with their expression levels as a combined input to the clustering system.

The approach clusters genes using a linear combination of two distances,  $d(g_i^{Exp}, g_j^{Exp})$  related to the gene expression data  $\Omega^{Exp}$  and  $d(g_i^{TF}, g_j^{TF})$  related to the regulation profiles  $\Omega^{TF}$ . Therefore, a weighting parameter  $\alpha$  ( $0 \leq \alpha \leq 1$ ) sets the balance between the gene distance in  $\Omega^{Exp}$  and in  $\Omega^{TF}$ :

$$d(g_i, g_j) = \alpha d(g_i^{Exp}, g_j^{Exp}) + (1 - \alpha) d(g_i^{TF}, g_j^{TF}) \quad (5-4)$$

In our implementation we compute this distance using the Pearson correlation distance measure (described in sect. 1.1), i.e.

$$d(g_i, g_j) = (1 - p(g_i, g_j)) \quad (5-5)$$

where  $p(g_i, g_j)$  is the Pearson correlation coefficient.

Using this distance metric we employed the traditional K-Means clustering algorithm, described in sect. 1.3.1 with different values of the parameter  $\alpha$ .

Analyzing the cluster obtained with this simple clustering approach we can see that the number of TFs with a high potential binding vales is much low compared to the entire set (of size  $m$ ). For this reason we propose an approach that at each iteration considers only the subset of relevant TF.

How to determine such subset is described in the following section.

**5.1.3 The Proposed Iterative Relational Clustering Approach**

The general relational clustering framework proposed in the last chapter, has been instantiated for this problem (visible in Fig. 5-2) where a loop between the relational learning and relational clustering components has been introduced.

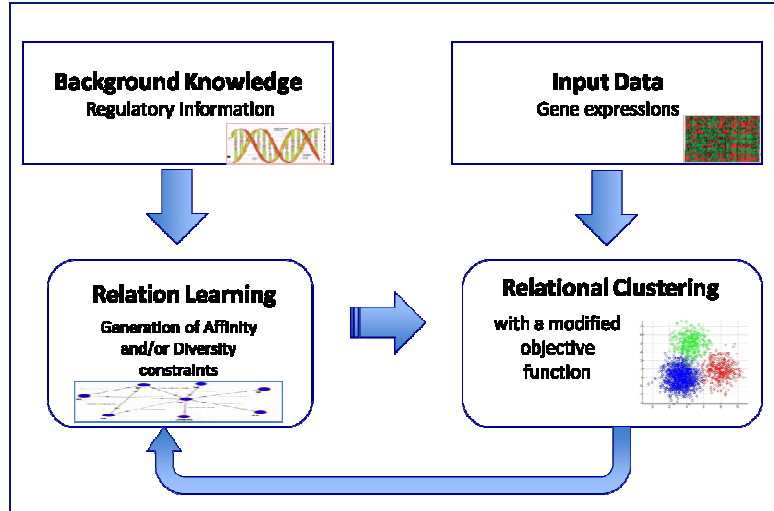


Fig. 5-2: Instantiation of general relational clustering framework

The initialization phase consists of clustering genes with respect to their expression levels, to obtain sets of genes that exhibit similar expression behaviors under various experimental conditions. In particular, we apply the K-means algorithm and the distance metric explained in subsection 2.1. Since we want to cluster genes with respect to their gene expression profile, we set the weighting parameter  $\alpha$ , presented in equation ( 5-4), equals to 1 in order to obtain the following distance measure:

$$d(g_i, g_j) = d(g_i^{Exp}, g_j^{Exp}) \quad (5-6)$$

where  $d(g_i, g_j)$  is computed using equation ( 5-5).

In order to build a *TF profile* for each gene belonging to the obtained clusters, we find the set of candidate TFs for each cluster of genes. Instead of considering all the TFs, which imply a profile dimension of  $m$  elements, we consider only the relevant ones.

In particular, we define a TF as relevant if it has a high potential binding factor with the promoter regions of genes belonging to that cluster.

We consider as candidate TFs only those TFs whose associated *p-value* (obtained by Pscan tool, as described above) is below a certain threshold that we set to  $10^{-4}$ , deemed as a significant level (Lee et al., 2002).

In this way, with all the binding values of chosen candidate TFs, we can build a new set  $\Omega^{TF}$  of transcriptional profiles of dimension less than  $m$ .

We now describe the main components of the framework depicted in Fig. 5-2 .

### 5.1.3.1 Relation Learning

In order to learn affinity and diversity relations, we perform a genes clustering with respect to gene TF profile using the traditional K-Means algorithm and a distance measure based on Pearson Correlation Coefficient.

Genes belonging to the same cluster  $C_c^{TF}$ , obtained by this step, will be considered as regulated by the same TFs and genes belonging to different clusters will be considered as having different transcriptional profiles.

The strength of genes relations depends on the distance between genes in the clustering model just obtained.

In order to include this relationships in the objective function of relational clustering phase, the two relation matrices  $R^A$  and  $R^D$  are defined; each element of  $R^A$ ,  $r_{ij}^A$ , represents the weight of an *Affinity Link* and the element  $r_{ij}^D$  of  $R^D$  represents the weight of an *Diversity Link*.

In particular,  $r_{ij}^A$  of matrix  $R^A$  is defined as the distance between the two transcriptional profiles  $g_i^{TF}$  and  $g_j^{TF}$ . Formally:

$$r_{ij}^A = \begin{cases} d(g_i^{TF}, g_j^{TF}) & \text{if gene } i \text{ and gene } j \in C_\alpha^{TF} \\ 0 & \text{otherwise} \end{cases} \quad (5-7)$$

Therefore, an *Affinity Link* between two genes  $i$  and  $j$  is defined as a link that invites us to put the  $g_i$  and  $g_j$  together in the same cluster during the subsequent phase, since this link means that the genes that we take into account have a high binding strengthens, and so a regulatory relationships, with the same TF(s).

A *Diversity Link* between two genes  $i$  and  $j$  suggests us not to put the two genes in the same cluster during the subsequent phase. In fact, since we want to find genes co-regulated by the same TF, a *Diversity* link says that genes  $i$  and genes  $j$  do not bind with the same TF. So, the element  $r_{ij}^D$  is represented by:

$$r_{ij}^D = \begin{cases} d(g_i^{TF}, g_j^{TF}) & \text{if gene } i \in C_\alpha^{TF} \text{ and gene } j \in C_\beta^{TF} \neq C_\alpha^{TF} \\ 0 & \text{otherwise} \end{cases} \quad (5-8)$$

### 5.1.3.2 Relational Clustering

The core of relational clustering step, depicted in Fig. 5-2, is based on refining the initial clustering model based only on gene expression profiles, in order to take into consideration regulatory relationships between genes based on their TF profile similarity.

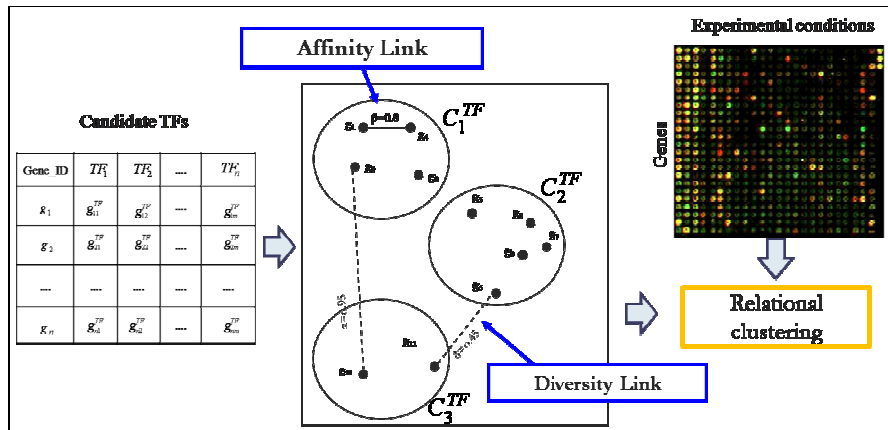


Fig. 5-3: Iterative process of learning distance measure and modify objective function

In particular, using the information matrixes  $R^A$  and  $R^D$  about gene *Affinity* and *Diversity* links and their respectively degrees of intensity, we perform a genes clustering with respect to gene expression profiles taking into account these learned regulatory relationships.

Also in this case, we use a K-Means clustering algorithm, but the objective function is modified (with respect to the traditional proposed in equation (

1-11)) by using both gene expressions data  $g_i^{Exp}$  and the *Affinity* and *Diversity* relationships.

$$\min \sum_{c=1}^c \left[ \sum_{i,j=1}^{|G|} \left[ d(g_i, g_j) a_{ic} a_{jc} - d(g_i, g_j) a_{ic} (1 - a_{jc}) + a_{ic} a_{jc} r_{ij}^D d(g_i, g_j) + a_{ic} (1 - a_{jc}) r_{ij}^A d(g_i, g_j) \right] \right] \quad (5-9)$$

In this way we minimize the sum of the distances between genes, expressed by their expression level measures, penalized by a function that considers regulatory and not regulatory links. In particular, if an *Affinity Link* (or a *Diversity Link*) is not preserved, the objective function will be penalized according to the weight  $r_{ij}^A$  (or  $r_{ij}^D$ ) learned from transcriptional profiles clustering.

The framework is iterative, in fact the new set of clusters is used to find a new set of relevant TFs that will be subsequently used to learn relations and apply the relational clustering. All these steps are repeated till the stability is reached i.e. when the set of relevant TFs doesn't change with respect to the previous iteration.

#### 5.1.4 Computational Experiments and Results

The iterative relational cluster algorithm presented in the above section has been evaluated by using two *Saccharomyces Cerevisiae* data sets: one (Gasch et al., 2000) consisting of 173 microarrays and 6172 genes, measuring the responses to various stress conditions; the other (Spellman et al., 1998) consisting of 77 microarrays and 6178 genes, measuring expression during cell cycle. These datasets are described more deeply in Appendix A1.1.

The promoter sequence data used by Pscan tool consists of the 500 base pairs upstream region of each gene. These sequences were retrieved from SGD (*Saccharomyces Genome Database*), an organized collection of genetic and molecular biological information about *Saccharomyces Cerevisiae* (Cherry et al., 1998).

Before applying the clustering procedure we perform a pre-processing phase with the aim of, not only remove as much as possible the systematic noise presented in microarray data, but also provide a basis for the next comparisons between genes. In particular, we perform a two-steps pre-processing strategy with missing value replacement and data normalization, and finally a preliminary genes selection to reduce the problem dimension.

The first step is based on data normalization and missing value replacement. Missing values often appear in gene expression data, due to various experimental limitations: technical reasons, like insufficient resolution, image corruption, or, simply, dust or scratches on the slide. The inability of clustering algorithms to handle such values necessitates their replacement.

In our approach for each gene, we substitute all missing values by the average of gene's expression profile.

Moreover, it is important to eliminate from microarray data variations due to non-biological factors. This procedure, known as normalization, is significant to obtain consistent data for following analysis. The normalization approach used in our work is the mean and SD normalization, whereby all microarray data are normalized so that every gene has been scaled to mean 0 and standard deviation of 1.

In the microarray datasets used for testing our approach thousands of gene expression levels were monitored. But, a considerable part of the data is related to genes that don't contribute to the underlying biological progress; in fact large numbers of genes exhibit nearly constant expression levels, as measured by the variance of the expression levels across arrays. Therefore we can discard these genes, and use only genes with high-variance of expression levels in the clustering process. In particular, after this selection, for our analysis we retained 1010 genes out of 6172 from (Gasch et al., 2000) dataset and 774 genes out of 6178 from (Spellman et al., 1998) dataset.

Evaluation of the results of our clustering algorithm necessitates careful consideration since there are no gold standards against which performance can be measured. In particular we evaluate our algorithm based on two points of view: first, we analyze the obtained modules based on co-expression and co-regulated patterns, then we compare the prediction ability of the SVTQ clustering approach with those of the proposed relational clustering method.

#### *Gene Expression and Gene Regulation Coherence*

Since the goal of the proposed method is to find modules of genes that are co-expressed and co-regulated, we analyze the modules obtained at each iteration.

For both *S. Cerevisiae* datasets, we initialized 50 clusters using standard k-means clustering algorithm (as proposed in (Segal et al., 2003)) and then we learn common regulators, by means of Pscan tool (Zambelli et al., 2009).

In particular, iterative relational clustering algorithm converges after 6 iterations for (Gasch et al., 2000) dataset with a set of 9 TFs, and after 5 iterations for (Spellman et al., 1998) dataset with a set of 8 TFs.

Main results are summarized, for (Gasch et al., 2000) dataset in Tab. 5-1 and Fig. 5-4 and for (Spellman et al., 1998) dataset in Tab. 5-2 and Fig. 5-5.

In these tables the number of co-expressed and co-regulated modules obtained at the different iterations is reported. In particular, to count the number of co-expressed clusters we computed for each group of genes the mean Pearson correlation coefficient and we considered only those clusters with a coefficient larger than 0.6. In Tab. 5-1 we reported, for different range values of Pearson correlation coefficient, the frequency of clusters in them.

A cluster is considered co-regulated if the Pearson correlation coefficient of the TF profile of its genes is larger than 0.7.

Analyzing these tables we can see that at iteration 0 only a small part of co-expressed clusters consists of co-regulated genes. The number of clusters that are both co-expressed and co-regulated increments with the iterations. In the last iteration we have that almost all co-expressed clusters are also co-regulated.

Examining Fig. 5-4 (Gasch et al., 2000), we can see that at iteration 0, when the clustering is applied only on gene expression data (the initialization of algorithm), we obtain only 10 clusters out of 50 that are both co-expressed and co-regulated. In the last iteration (iteration 6), we obtain 29 out of 31 both co-expressed and co-regulated clusters. Genes which do not belong to these clusters may be considered as genes that are not involved in the regulation process.

	ITERATION 0		ITERATION 1		ITERATION 2		ITERATION 3		ITERATION 4		ITERATION 5		ITERATION 6	
	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg
Pearson Correlation Threshold														
0.6<Corr<0.7	7	3	4	3	7	4	3	5	4	4	6	5	7	6
0.7<Corr<0.8	8	3	12	8	12	9	15	11	16	12	15	14	14	14
0.8<Corr<0.9	12	3	8	5	6	4	4	4	6	5	7	6	8	7
Corr>0.9	2	1	1	1	2	2	2	2	2	2	2	2	2	2
Total cluster number	29	10	25	17	27	19	24	22	28	23	30	27	31	29

Tab. 5-1: Iterative relational clustering algorithm results on (Gasch et al., 2000) dataset

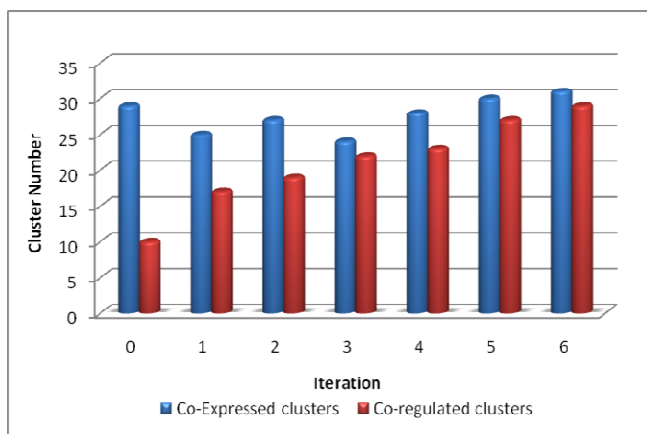
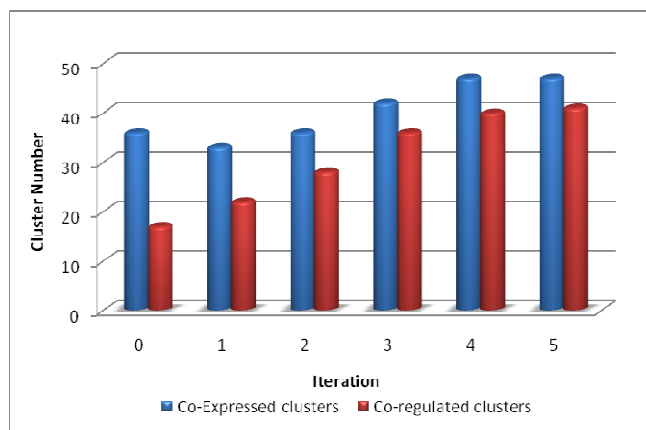


Fig. 5-4: Number of co-expressed and co-regulated clusters for (Gasch et al., 2000) dataset

Also the results obtained on the dataset of (Spellman et al., 1998) emphasized the ability of our approach to identify modules of co-expressed and co-regulated genes. Also in this case, as depicted in Fig. 5-5, from iteration 0, where only 17 clusters are co-expressed and co-regulated, we obtain at iteration 5, 41 clusters of genes both co-expressed and co-regulated.

	ITERATION 0		ITERATION 1		ITERATION 2		ITERATION 3		ITERATION 4		ITERATION 5	
	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg	# Co-exp	# Co-exp / Co-reg
Pearson Correlation Threshold												
0.6<Corr<0.7	11	5	9	7	9	8	12	10	11	11	11	11
0.7<Corr<0.8	15	9	16	12	18	15	17	16	18	17	18	17
0.8<Corr<0.9	9	3	7	3	8	5	12	9	16	11	16	11
Corr>0.9	1	0	1	0	1	0	1	1	2	1	2	2
Total cluster number	36	17	33	22	36	28	42	36	47	40	47	41

Tab. 5-2: Iterative relational clustering algorithm results on (Spellman et al., 1998) dataset



**Fig. 5-5:** Number of co-expressed and co-regulated clusters for (Spellman et al., 1998) dataset

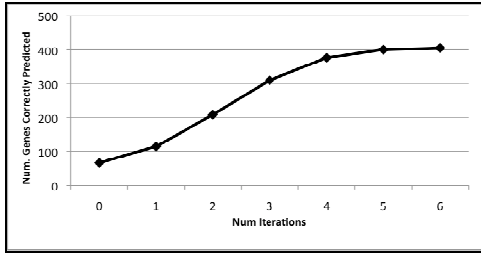
### *Predicting Expression from Transcriptional Profiles*

To assess the consistency of cluster analysis in the inference of co-regulation, we also evaluated our iterative relational clustering algorithm results by computing the number of genes whose cluster assignment is correctly predicted based on transcription factors profiles. Specifically, with the clustering assignment obtained from the *relational clustering phase*, we built Naive Bayes classification models and compared the cluster assignment of each gene when we consider only the transcriptional data to its cluster assignment considering both expression and transcriptional data.

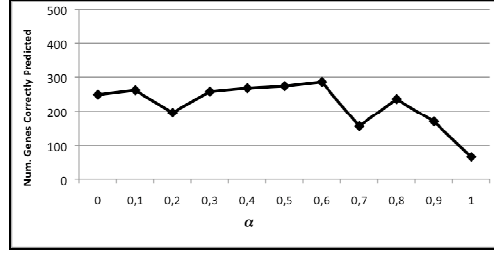
Fig. 5-6 and Fig. 5-8 show, for both datasets, the total number of genes whose expression-based cluster assignment is correctly predicted using only the transcriptional data, as the algorithm progresses.

To verify our approach we also compared our prediction results with those obtained using the simple clustering approach that follows the idea proposed by (Clements et al., 2007), called also STVQ in (Graepel et al., 1998). In these approach, like just said before, is used a parameter  $\alpha$  ( $0 \leq \alpha \leq 1$ ) as a weighting parameter that sets the balance between the expression distance and the regulatory distance. Clearly, when  $\alpha = 1$  the simple clustering algorithm is performed only on gene expression data and, therefore, when  $\alpha = 0$  clustering algorithm is performed only on transcriptional data.

The number of genes correctly predicted by the initial iteration of our approach must be equivalent to the number of genes calculated by traditional approach using an alpha equal to 1, in fact in both cases clustering is based only on gene expression data.

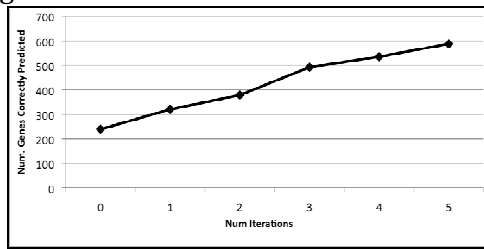


**Fig. 5-6:** Genes correctly predicted by iterative relational clustering approach for (Gasch et al., 2000) dataset.

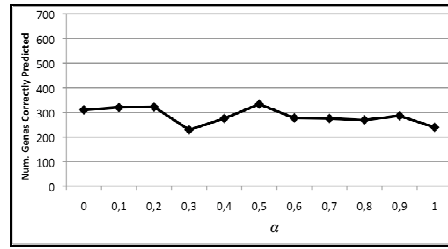


**Fig. 5-7:** Genes correctly predicted by STVQ approach for (Gasch et al., 2000) dataset.

As can be seen in figures Fig. 5-6, Fig. 5-7, Fig. 5-8 and Fig. 5-9, the predictions get better across the learning iterations, and the results obtained in the last iteration significantly outperform the simple approach with the best  $\alpha$  ( $\alpha=0,6$ ). In particular, our model converges to 405 genes correctly predicted in the stress dataset (Gasch et al., 2000), respectively, compared to 296 for the simple clustering approach. Also analyzing results obtained on (Spellman et al., 1998) dataset, it's possible to see that our relational clustering algorithm at the last iteration, predicts 587 genes against the 333 genes of the simple clustering algorithm.



**Fig. 5-8:** Genes correctly predicted by iterative relational clustering approach for (Spellman et al., 1998) dataset



**Fig. 5-9:** Genes correctly predicted by STVQ approach for (Gasch et al., 2000) dataset.

Finally, we analyzed also the set of candidate TFs obtained at each iteration of our iterative relational clustering approach. In tables below (Tab. 5-3 and Tab. 5-4) we can see that, for both datasets, iteration after iteration the number of TF is reduced until it converges to a stable set of TFs.

TRANSCRIPTION FACTORS						
ITERATION 0	ITERATION 1	ITERATION 2	ITERATION 3	ITERATION 4	ITERATION 5	ITERATION 6
GAL4	GAL4	ABF1	ABF1	ABF1	ABF1	ABF1
MATHALPHA2	GCN4	GAL4	GAL4	GAL4	GAL4	GAL4
MCM1	MCM1	MCM1	MCM1	MCM1	MCM1	MCM1
MIG1	MIG1	MIG1	MIG1	MIG1	MIG1	MIG1
PDR1/PDR3	PDR1/PDR3	PDR1/PDR3	PDR1/PDR3	PDR1/PDR3	PDR1/PDR3	PDR1/PDR3
PHO4	PHO4	PHO2	PHO4	PHO4	PHO4	PHO4
RAP1	RAP1	RAP1	RAP1	RAP1	RAP1	RAP1
REPRESSOR OF CAR1	REPRESSOR OF CAR1	REPRESSOR OF CAR1	REPRESSOR OF CAR1	REPRESSOR OF CAR1	REPRESSOR OF CAR1	REPRESSOR OF CAR1
RLM1	RLM1	RLM1	SCB	ROX1	SCB	SCB
ROX1	SMP1	SCB	STE12			
SWI5	SWI5	SWI5				
TBP						

**Tab. 5-3:** Transcription factors found by our iterative relational clustering algorithm on (Gasch et al., 2000) dataset



TRANSCRIPTION FACTORS					
<i>ITERATION</i> <i>0</i>	<i>ITERATION</i> <i>1</i>	<i>ITERATION</i> <i>2</i>	<i>ITERATION</i> <i>3</i>	<i>ITERATION</i> <i>4</i>	<i>ITERATION</i> <i>5</i>
GAL4	MATHALPHA2	MATHALPHA2	MCB	MCB	MCB
MATHALPHA2	MCB	MCB	MCM1	MCM1	MCM1
MCB	MCM1	MCM1	PDR1/PDR3	PDR1/PDR3	PDR1/PDR3
MCM1	PDR1/PDR3	PDR1/PDR3	REPRESSOR OF CAR1	REPRESSOR OF CAR1	REPRESSOR OF CAR1
PDR1/PDR3	RAP1	RAP1	RLM1	ROX1	ROX1
RAP1	REPRESSOR OF CAR1	REPRESSOR OF CAR1	ROX1	SCB	SCB
REPRESSOR OF CAR1	RLM1	RLM1	SCB	SMP1	SMP1
RLM1	ROX1	ROX1	SMP1	STE12	STE12
ROX1	SCB	SCB	STE12		
SCB	SMP1	SMP1			
SMP1	STE12	STE12			
STE12	SWI5	SWI5			
SWI5	UASPHR	XBP1			
UASPHR	XBP1				
XBP1					

**Tab. 5-4:** Transcription factors found by our iterative relational clustering algorithm applied on (Spellman et al., 1998) dataset

In conclusion, in this application, we proposed an iterative relational clustering approach that integrates information concerning known transcription factors-gene interactions with gene expression data and whose goal is to identify transcriptional modules of co-regulated genes, i.e. group of genes that are not only expressed similarly under the measured conditions, but also share a common regulatory program.

Obtained results indicate that the proposed method discovers iteration after iteration, modules of genes that are both highly coherent in their gene expression and regulatory profiles. A comparison to the common approach of constructing clusters based only on a linear combination of gene expression profiles and regulatory profiles, shows that the prediction of gene expression from transcriptional profiles of our method improve across the different iterations and outperforms the common approach.

## ***5.2 Detecting the Most Effective Cancer Drug: NCI-60 Dataset***

The ultimate goal of this case study is to define a model which, given the gene expression profile related to a specific tumour tissue, could help in selecting a set of most responsive drugs.

In this Thesis this is accomplished through the instantiation of the general relational framework presented in sect.4.4.

Main results of this analysis are illustrated in (Fersini et al., 2009a; Fersini et al., 2009b).

This approach groups cell lines using drug response information and taking into account cell-to-cell relationships derived from their gene expression profiles.

Next subsections provide an overview of the pharmacogenomics problem, a review of the state of the art and a deeply description of the relational framework.

### **5.2.1 The Pharmacogenomics Problem: State of the Art**

Microarray technologies (as for example cDNA microarrays and affymetrix chips), that measure the expression level of thousands of genes simultaneously, have steady established themselves as a standard tool in biological and biomedical research. As already underlined in the previous sections, thanks to these recent progresses large amount of data have been collected, offering important opportunities to increase the knowledge related to complex biological phenomena.

One of the most challenging problem in biomedical research is related to the discovery of embedded relationships among human cancer, gene expression profile and drug activity. Highlighting these relationships is of crucial importance for several objectives, among others: identification of mechanisms of the cancer development, design of new molecular targets for anti-cancer drugs and definition of an individual therapy driven by a specific gene profile.

Several studies tried to integrate gene expression data with drug-response profiles in a sequential manner.

A first gene-drug integrative analysis was presented in (Paull et al., 1989). Authors developed a tool, named "COMPARE", able to show that the growth inhibitory patterns against different cancer cell lines are well correlated with the mechanism of action of anticancer therapy.

One of the most relevant alternative studies into the pharmacology of cancer relates to (Scherf et al., 2000), in which a hierarchical clustering algorithm, with several similarity metrics, has been used to analyze: (1) cell-to-cell correlation on the basis of gene expression and drug activity profiles, (2) relationships between drug activity patterns and mechanism of action, (3) gene-drug correlation on the basis of gene expression and drug activity profiles. In (Chang et al., 2002) and (Chang et al., 2005) the relationships between gene expression profiles and drug responses are investigated by both unsupervised and supervised machine learning algorithms. In particular, while through the unsupervised Soft Topographic Vector Quantization (STVQ) algorithm (Graepel et al., 1998) (Clements et al., 2007) authors shown that gene expression profiles are more related to the kind of cancer than to drug activity patterns, through the

supervised Bayesian networks (Chang et al., 2002) and (Chang et al., 2005), some biologically meaningful relationships among gene expression levels, drug activities, and cancer types have been revealed.

An alternative approach to the traditional clustering and classification methods for discovering relationships among genes and drugs across different cell lines is represented by the “structure driven” approaches and in particular by Biclustering algorithms (subsection 3.1). An interesting investigation has been proposed in (Kutalik et al., 2008) that applied the well known Iterative Signature Algorithm for uncovering co-modules, i.e. smaller building blocks that exhibit similar patterns across certain genes and drugs in some of the cell lines.

After a deeply analysis of the results of the above quoted papers and of a wide set of related approaches, we can highlight an interesting remark: drug activity patterns are less related to the organ of origin compared to the gene expression profile. This suggests us that a gene expression profile of a cell line plays a fundamental role, independently from the tissue of origin, to understand anticancer therapy responses.

Inspired by this remark we perform different cluster analysis aimed at linking gene expression profiles to drug activity patterns.

In particular, initially we apply traditional clustering algorithms for a first analysis of the problem and then we apply the relational clustering framework presented in subsection 4.4 aimed at investigating whether drug response can be related to subsets of genes.

Finally, we exploit the output of cluster analysis to induce a specific Bayesian Network able to predict the response of a set of drugs.

### 5.2.2 Traditional Approaches: K-Means and SVT Algorithms

The NCI60 dataset, described in Appendix A and presented in (Scherf et al., 2000), provides a suite of comprehensive measurements on a set of cell lines derived from 9 kinds of cancers: colorectal, renal, ovarian, breast, prostate, lung and central nervous system origin, as well as leukaemia and melanoma.

The dataset can be viewed as a set into the space  $R^{n+m}$  defined as:

$$\Omega = \{c | c = (c^G, c^D), c^G \in R^m, c^D \in R^n\} \quad (5-10)$$

where  $c$  is a cell line,  $c^G$  represents the gene expression level into a space  $R^m$  and  $c^D$  denotes the drug response into a space  $R^n$ . In particular,  $c^G$  has been derived by using the cDNA microarray and  $c^D$  by assessing the grown inhibition activities (GI50) after 48 hours of drug treatment through Sulphorhodamine B.

We can consequently define  $\Omega^G$  and  $\Omega^D$  as the set of the cell lines represented through their gene expression profiles and their drug activity response respectively:

$$\Omega^G = \{c^G | c = (c^G, c^D), c \in \Omega\} \quad (5-11)$$

$$\Omega^D = \{c^D | c = (c^G, c^D), c \in \Omega\} \quad (5-12)$$

Even in this case we used as distance measure based on the *Pearson Correlation* ( $p(c_i, c_j)$ ), already described in section 1.1, i.e.

$$d(c_i, c_j) = 1 - p(c_i, c_j) \quad (5-13)$$

where  $c_i$  and  $c_j$  represent two cell lines.

A first analysis of the problem is done investigating two traditional clustering algorithms: the traditional K-means and the STVQ. Both these algorithms use a “flat” representation of data, i.e. by representing each cell line as a vector in  $R^{n+m}$ , like in Fig. 5-10.



Fig. 5-10: General “flat” representation of cell lines

The goal is to minimize the error that occurs by assigning a cell line to a given cluster through a distance measure that considers the entire space  $R^{n+m}$ . Even if both K-means and STVQ algorithms use the same data representation, STVQ, like already explained in the previous case study, uses a distance measure based on a linear combination of gene expression profile and drug activity pattern distances tuned by the parameter  $\alpha$ , i.e.

$$d(c_i, c_j) = \alpha d(c_i^g, c_j^g) + (1 - \alpha) d(c_i^d, c_j^d) \quad (5-14)$$

During the experimental investigation we have used three different values:  $\alpha=0$ ,  $\alpha = 0.5$  and  $\alpha= 1$ . Each of these values has been used in order to produce a partitioning solution that considers only the distance based on gene expression profiles ( $\alpha=0$ ), the distance based on gene expression profiles and drug activity patterns ( $\alpha=0.5$ ) and, finally, the distance based only on drug activity patterns ( $\alpha=1$ ).

### 5.2.3 The Proposed Relational Clustering Approach

The instantiation of the general relational clustering framework for this case study is depicted in Fig. 5-12. In particular in the “relation learning” component we learn relationships between cell lines over the gene space, while in the “relational clustering” component phase we incorporate these relationships along with an underlying objective function over the drug space.

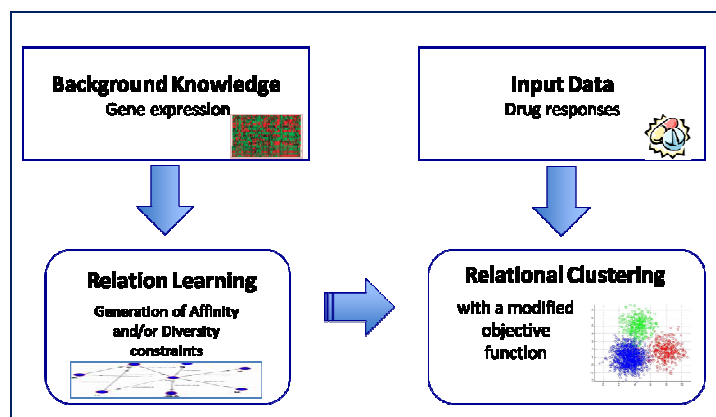


Fig. 5-11: Instantiation of the general relational clustering framework

### 5.2.3.1 Relation Learning

A relation among two instances can be either an *Affinity* or a *Diversity* link where with *Affinity link* we want to describe the relation between two instances that shared similar expression profile and with *Diversity link* we want to underline the fact that the two instances don't share a common expression profile.

Like we have described in the general relational clustering framework we assign to each relation a different degree of intensity.

Since no relations between cell lines in the gene space are given in advance, we can discover them by formulating a clustering problem.

Given the elements  $c_i^G \in \Omega^G$ , a set of clusters  $C_j$  with  $j = 1 : J$ , the clustering problem can be defined as in equation ( 1-11), where each element  $c_i^G$  is assigned to a cluster  $C_j^G$  such that the intra-cluster distance is minimized and the inter-cluster distance is maximized. This issue has been addressed through the k-Means clustering algorithm described in previous subsection.

The obtained set of clusters leads us to define two matrices of relationships  $R^D$  and  $R^A$  that will be used in the subsequent objective function optimization over the drug space.

$R^D$  is a  $|\Omega|X|\Omega|$  matrix whose elements  $r_{ik}^D$  represents the weights of the *Diversity links* between elements belonging to different clusters and will suggest in the following phase that two cell lines should not be placed in the same module.

More formally,  $r_{ik}^D$  is defined as the distance (computed as in ( 5-10)) between  $c_i^G$  and  $c_k^G$  i.e.

$$r_{ik}^D = \begin{cases} d(c_i^G, c_k^G) & \text{if } c_i^G \in C_\alpha^G \text{ and } c_k^G \in C_\beta^G \neq C_\alpha^G \\ 0 & \text{otherwise} \end{cases} \quad (5-15)$$

The matrix  $R^A$ , having the same dimension of  $R^D$ , represents the weight of the *Affinity links* of elements belonging to the same cluster and will suggest in the following phase that two elements should be placed in the same module.

The element  $r_{ik}^A$  is given by

$$r_{ik}^A = \begin{cases} d(c_i^G, c_k^G) & \text{if } c_i^G \wedge c_k^G \in C_\alpha^G \\ 0 & \text{otherwise} \end{cases} \quad (5-16)$$

### 5.2.3.2 Relational clustering

The second phase of the computational process is focused on grouping cell lines using drug response information while taking into account cell to-cell relationships coming from the previous stage. In particular, this clustering step is aimed at minimizing the sum of the distances between elements, expressed by their drug activity response, penalized by a function that takes into account affinity and diversity links.

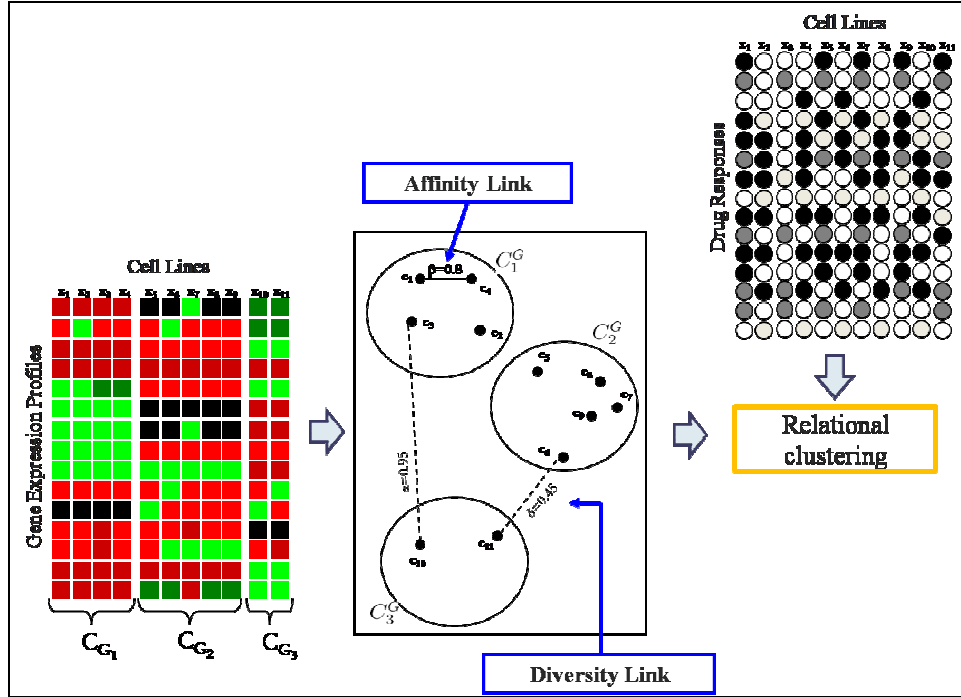


Fig. 5-12: Computational process of the proposed relational clustering framework

Let  $c_i^D \in \Omega^D$  be a given cell line represented by its drug response features and  $C_j^D$ , with  $j = 1 : J$ , be a set of clusters. Defining  $Z$  as in equation ( 1-10 ), for cell lines expressed by their drug responses, i.e.

$$z_{ij} = \begin{cases} 1 & \text{if } c_i^D \in C_j^D \\ 0 & \text{otherwise} \end{cases} \quad (5-17)$$

The problem can be formulated as follows:

$$\min \sum_{j=1}^J \left[ \sum_{ik=1}^{|\Omega|} \left[ d(c_i^D, c_k^D) z_{ij} z_{kj} - d(c_i^D, c_k^D) z_{ij} (1 - z_{kj}) + z_{ij} z_{kj} r_{ik}^D d(c_i^D, c_k^D) + z_{ij} (1 - z_{kj}) r_{ik}^A d(c_i^D, c_k^D) \right] \right] \quad (5-18)$$

The optimization problem reported in equation ( 5-18), can be solved through an adaptation of K-means heuristic reported before.

In this way if an *Affinity link* (or a *Diversity link*) is not preserved, the objective function is penalized according to the weight  $r_{ik}^A$  (or  $r_{ik}^D$ ) and the distance between cell lines  $i$  and  $k$  represented in terms of their drug responses features,  $d(c_i^D, c_k^D)$ .

## 5.2.4 Computational Experiment and Results

In order to obtain a meaningful representation of the NCI60 data, in terms of discriminative features that can be used by the proposed machine learning algorithm, two pre-processing steps have been performed:

1. *Cell lines with strong variation and missing values.*

We defined  $\Omega_1$  by representing each cell line into the gene expression and drug activity spaces  $R^m=1376$  and  $R^n=1400$  respectively. With respect to  $R^m$ , genes have been selected from the original NCI60 dataset (characterized by

9073 genes) in order to have 5 or fewer missing values and to show strong pattern of variation among the 60 cell lines. This representation corresponds to the NCI60 *T-matrix*.

With respect to  $R^n$ , we maintained the 1400 drugs stated into the original dataset, where each compound has been tested one at time and independently. This dataset representation, corresponding to the NCI60 *A-matrix*, need a further pre-processing activity aimed at dealing with missing values: for each gene (or drug) that show one or more missing values, its average gene expression value (or its average drug activity) over the 60 cell lines is used as replacement.

A description of both *T-matrix* and *A-matrix* is in Appendix A.

## 2. Cell lines with strong variation and no missing values.

We defined  $\Omega_2$  by removing from the original dataset (composed by *T-matrix* and *A-matrix*) those genes and drugs for which at least one cell line had a missing value. In this case, each cell line is represented into the gene expression and drug activity spaces  $R^m = 555$  and  $R^n = 836$  respectively.

In both reduced datasets  $\Omega_1$  and  $\Omega_2$  cell lines have been normalized in order to have mean equal to 0.

In order to evaluate the quality of the proposed relational clustering algorithm, we used similarity-oriented evaluation measures as the Average Pearson Correlation Coefficient, as well as classification-oriented measures as F-Measure and Entropy. An overview of these clustering evaluation measures has been already presented in subsection 1.4.

In particular, for similarity-oriented evaluation measure, we computed the widely adopted in biology studies, *average Pearson Correlation Coefficient* defined as:

$$\bar{P} = \sum_{j=1}^J \frac{m_j}{|\Omega|} \left[ \frac{2}{n_j(n_j - 1)} \sum_{i < k} \text{corr}_{i,k} z_{ij} z_{kj} \right] \quad (5-19)$$

where  $J$  is the number of clusters and  $m_j$  is the cardinality of the cluster obtained by our relational clustering process. Here we use  $z_{ij}$  to say that the cell line  $i$  has been assigned to the cluster  $j$  in the overall clustering process.

We estimate  $\bar{P}$  in two ways: in one case  $\bar{P}$  is computed considering the correlation between instances  $i$  and  $k$  represented by their gene expression profiles and in another case using their drug response profiles. Following this evaluation we have a scalar value  $\bar{P}^g$  related to the overall correlation of the obtained clusters, with respect to gene expression profiles and  $\bar{P}^d$  related to drug activity patterns.

With respect to the traditional classification-oriented evaluation measure, we use *F-Measure* which combines the Precision and Recall measure typical of Information Retrieval.

In particular, if we apply to this problem the general definition introduced in subsection 1.4.1.1, we obtain the following formulation.

Given a set of class label  $L$  representing the type of cancer (in this case equals to 9), we compute the *Precision* and *Recall* for each class label  $l \in L$  with respect to the cluster  $j$  as:

$$Precision(l,j) = \frac{m_{lj}}{m_i} \quad (5-20)$$

$$Recall(l,j) = \frac{m_{lj}}{m_j} \quad (5-21)$$

Where:

- $m_{lj}$  is the number of cell lines belonging to the class label  $j$  and located in the cluster  $i$
- $m_j$  represents the cardinality of cluster  $j$
- $m_l$  is the number of cell lines with class label  $l$ .

The F-Measure  $F(l, j)$  for each class label  $l \in L$  is computed as the harmonic mean of Precision and Recall, like in equation ( 1-23).

The overall quality of the obtained clustering solution is given by a scalar  $F^*$  computed as the weighted sum of the F-Measure values taken over all the class labels  $j \in L$ :

$$F^* = \sum_{l=1}^L \frac{m_l}{|\Omega|} \max_{j \in J} \{F(l, j)\} \quad (5-22)$$

The other classification-oriented measure used for evaluating clustering output is the Entropy Measure that, like already explained, evaluates the purity of the clusters with respect to the given class labels.

To compute the total entropy  $E^*$  of a set of obtained cluster we have used the same formulation of equation ( 1-18), and in particular:

$$E^* = \sum_{j=1}^J \frac{m_{lj}}{|\Omega|} E_j \quad (5-23)$$

where  $m_{lj}$  is the number of elements belonging to the class label  $l$  and located in the cluster  $j$ . In Tab. 5-5 and in Tab. 5-6 we report a performance comparison, over  $\Omega_1$  and  $\Omega_2$  respectively, among the results of our relational clustering approach (RC), the traditional k-Means (KM) and Soft Topographic Vector Quantization (STVQ) algorithm.

In particular, for STVQ algorithms, we report results obtained for the three different values of the tuning parameter. Since all the evaluated algorithms depend on the initial choice of the representative element of each cluster (centroid), we show the average performance obtained over 1000 runs.

		$p^G$	$p^D$	$F^*$	$E^*$
K-Means		0.4373	0.8032	0.5407	1.010
STVQ	$\alpha=0$	<b>0.4996</b>	0.8210	0.5481	1.0334
	$\alpha=0.5$	0.4801	0.8311	0.5005	1.0871
	$\alpha=1$	0.3606	<b>0.8330</b>	0.4320	1.2610
Relational Clustering		0.4873	0.8231	<b>0.5591</b>	<b>0.9868</b>

**Tab. 5-5:** Computational results on  $\Omega_1$

		$p^G$	$p^D$	$F^*$	$E^*$
K-Means		0.5147	0.8748	0.5231	1.0527
STVQ	$\alpha=0$	<b>0.5573</b>	0.8646	0.5455	1.0233
	$\alpha=0.5$	0.5394	0.8700	0.5307	1.0613
	$\alpha=1$	0.4430	<b>0.8762</b>	0.5455	1.388
Relational Clustering		0.5436	0.8665	<b>0.5619</b>	<b>0.9684</b>

**Tab. 5-6:** Computational results on  $\Omega_2$



Results have shown that all the tested algorithms produces better performance on the reduced dataset  $\Omega_2$  than on the dataset  $\Omega_1$ .

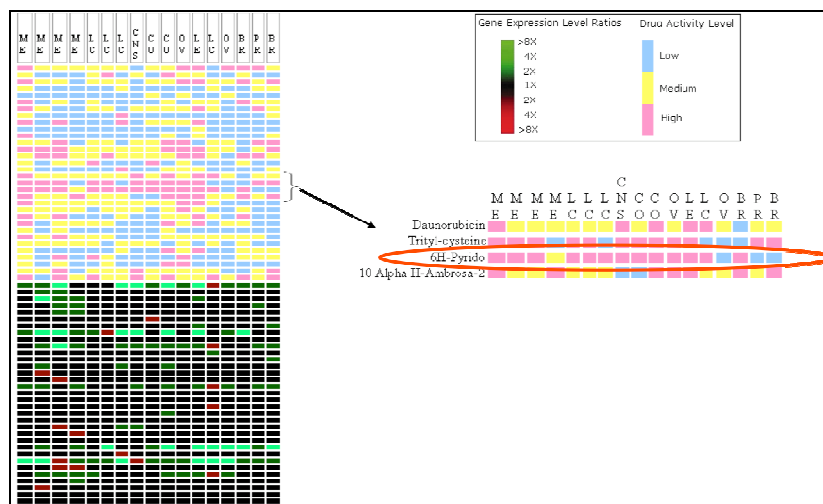
It is interesting to note that, even if we set the number of clusters  $J$  in the proposed relational clustering algorithm equal to 9, in some cases our algorithm converges, due to the force of *Affinity* and *Diversity* links, to a solution that provides an inferior number of clusters.

It's clear that the best results with respect to  $P^D$  and  $P^G$  are obtained by STVQ approach with  $\alpha = 0$  and  $\alpha = 1$ , albeit our approach is very close to these correlation values implying that the obtained groups of cell lines are homogeneous both from gene expression profile and drug activity response point of view. In this way, cell lines will likely respond similarly to the set of considered compounds thanks to their high drug and gene correlations. Moreover, with respect to the classification-oriented quality measures, our approach produces a clustering solution that is characterized by more purity and discriminative power than the traditional technique.

From a biomedicine point of view, the model defined by our approach could help, given the gene expression profile related to a cell line  $c_T$ , in selecting the set of most responsive drugs. In order to suggest these compounds, the cell line  $c_T$  could be associated to a specific module by using the minimum distance between  $c_T$  and the representative element of each obtained cluster.

Having the assignment of  $c_T$  to a given module, the active drugs could be ranked considering the number of samples having high response.

Therefore the most active drugs are ranked considering the number of samples having high response, which belongs to the cluster in which the new instance is assigned. An interesting sub-sample is depicted in Fig. 5-13.



**Fig. 5-13:** A sub-sample of the obtained clustering solution and a particular sub-pattern representing an example of the active drugs.

The cell lines are depicted in terms of their drug response and subsequently their gene expression profile. Each drug has been represented in terms of high, medium and low level of response with respect to its range of activity variation. The gene profile of each cell line is coloured in order to reject the mean-adjusted expression level of the gene (row) and cell line (column). In this way we can

associate to a given cell line the set of most responsive drugs, ranked by a simple frequentist intra-cluster approach.

*Bayesian Networks for Predicting Drug Response*

This simple frequentist ranking approach for the suggestion of the most responsive drugs do not take into account causal dependencies between the expression level of genes and the activity level of drugs.

A more sophisticated approach able to consider these causal relationships is represented by Bayesian Networks, a probabilistic graphical model that compactly represent the joint probability distribution of  $M$  random variables,  $Y=\{Y^1, Y^2, \dots, Y^M\}$ .

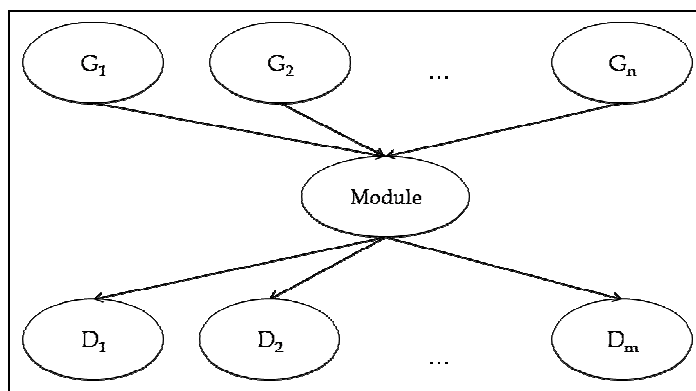
The main assumption, captured graphically by a dependency structure, is that each variable is directly influenced by only few others. A probability distribution is represented as a directed acyclic graph (DAG) whose nodes represent random variables and whose edges denote direct dependencies (causal relationships) between a node  $Y$  and its set of parents  $Pa(Y)$ .

Formally, a Bayesian Network asserts that each node is conditional independent of its non-descendants given its parents. This conditionally independence assumption allows us to represent concisely the joint probability distribution.

If we consider a distribution  $P$  over  $M$  feature, it can be decomposed as the product of  $M$  conditional distributions:

$$P(y^1, y^2, \dots, y^M) = \prod_{j=1}^M P(y^j | y^1, y^2, \dots, y^{j-1}) = \prod_{j=1}^M P(y^j | Pa(y^j)) \quad (5-24)$$

where  $P(y^j | Pa(y^j))$  is described by a conditional probability distribution (CPD).



**Fig. 5-14:** Bayesian Networks for modelling the NCI60 dataset

Fig. 5-14 shows our instantiation of Bayesian Networks for gene-drug dependency analysis. Here, the DAG denotes the dependency structure among gene expression, modules and drug activities. We could gain an insight that the expression pattern of genes influences the activity level of drugs through the module assignment. This structure of Bayesian Networks has been defined for inducing a probabilistic model able to predict the drug response of a new cell, only by providing its gene expression profile.

In order to evaluate the predictive power of the defined Bayesian Network, we performed a leave-one-out cross validation. This means that we use a single cell line from the original dataset as the validation data and the remaining cell lines as the training samples. CPDs of the defined network are derived by specifying

for each cell line not only its gene expression profile and pattern of drug responses, but also its associated module generated as output of the clustering process.

		Correctly Predicted Drugs
K-Means		11957
STVQ	$\alpha=0$	11782
	$\alpha=0.5$	11964
	$\alpha=1$	12009
Relational Clustering		<b>12292</b>

**Tab. 5-7:** Computational results of Bayesian networks on  $\Omega_2$

In Tab. 5-5 we reported the results on inducing BNs on the set of cell lines belonging to  $\Omega_2$ , according to the output obtained by the three clustering algorithms (average over the 1000 clustering runs). This analysis has been limited only to the set  $\Omega_2$  due to the presence of “replaced” missing values in  $\Omega_1$ . The quality of prediction of each induced network has been evaluated by counting the total amount of drug responses that, along the entire 60 cell lines, are correctly inferred. As highlighted in Tab. 5-8, the BN that obtain the quite encoring result is the one trained with the modules defined by the proposed relational clustering algorithm.

In conclusion, in this application we have instantiated the relational clustering approach to cluster the 60 cell lines of the NCI60 dataset for defining groups of cell lines by using drug response information and taking into account cell-to-cell relationships defined by the similarity of their gene expression profiles.

At the end of cluster analysis Bayesian Networks have been instantiated for inducing a probabilistic model able to predict drug responses of a new tumour tissue.

The experimental results show that the proposed method outperforms the traditional distance-based techniques. In particular, our clustering algorithm brings to the definition of clusters that are homogeneous both in terms of gene expression and drug activity profiles. Moreover, our approach produces a clustering solution that is characterized by more purity and discriminative power than the traditional technique.

### **5.3 Oral Anticoagulation Therapy**

In this section we present a clinical application of the general relational clustering framework.

Results obtained in this third case study are synthesized in (Archetti et al., 2009d; Archetti et al., 2009e).

The application's aim is focused on grouping patients undergoing Oral Anticoagulation Therapy in order to profiling them based on their behavior and clinical features and consequently suggest physicians the correct drug dosage.

In this way, with the instantiation of the relational clustering framework presented in section 4.4, we want to cluster patients using clinical and genotypic information, taking into account patients' relationships derived from their behavioural data.

For this application we used a proprietary dataset, built in collaboration with two hospitals in Milan: Istituto Auxologico and Clinica Humanitas. A description of these data is presented in Appendix A.

For each patient we have collected clinical data (like gender, age and OAT data), therapeutical data (concomitant drugs) and visit data (a time series of INR and doses measurements).

In this case study, an initial pre-processing phase of the collected data and a preliminary study of the principal patients' characteristics has been performed before applying the relational clustering framework.

In the next subsections a description of the clinical problem and in particular of Oral Anticoagulation Therapy is provided; subsequently are presented a review of the state of the art, an initial patient's characterization and a description of the relational clustering framework.

#### **5.3.1 The Clinical Problem: State of the Art**

Oral anticoagulation therapy (OAT), largely performed by warfarin-based drugs, is commonly used for patients with a high risk of blood clotting which can lead to stroke or thrombosis. A lot of persons start taking warfarin each year; physicians commonly prescribe it for patients with a history of atrial fibrillation, recurrent stroke, deep vein thrombosis, or pulmonary embolism, as well as for patients who have had heart valve replacements.

A major challenge in treating patients in OAT is that the optimal dose varies greatly from person to person. Further, if the dose taken is too high, users are subject to increased risk of serious bleeding. Finally, if the dose is too low, users are subject to increased risk of stroke.

Currently, the state of the patient, with respect to anticoagulation, is captured by the index INR (International Rationalised Ratio), which is to be kept within a therapeutic range defined by physician during the first visit. A representation of the risks concerned to INR index is illustrated in Fig. 5-15.

In particular, if INR value is over the assigned range then hemorrhagic risk increments, while if INR value is under the range then thrombotic risk grows. In this way, the appropriate dose is determined by monitoring INR index and altering the dose if INR index is out of range.

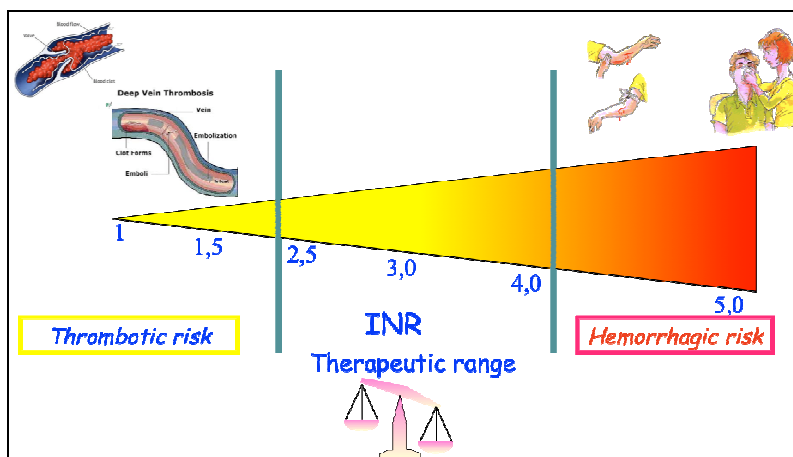


Fig. 5-15: Therapeutic INR range

The OAT workflow is rather complex (as shown in Fig. 5-16) and demanding given the typically old age of most patients.

It is therefore a heavy therapeutic modality for patients and often it is not accepted by physicians for different reasons like: potential hemorrhagic or thrombotic risk during the therapy, difficulties in management and communication between laboratory and patient (probably due to patients' age). In addition, patient's response to therapy is influenced by several other factors, including genetic factors, as evidenced by recent scientific papers, dietary habits and intake of other drugs that interact in a complex and difficult to predict way with INR.

In conclusion, frequent sampling (at least once in 2-3 weeks) of the INR and careful dosage adjustments are needed for the INR to stay within its assigned range.

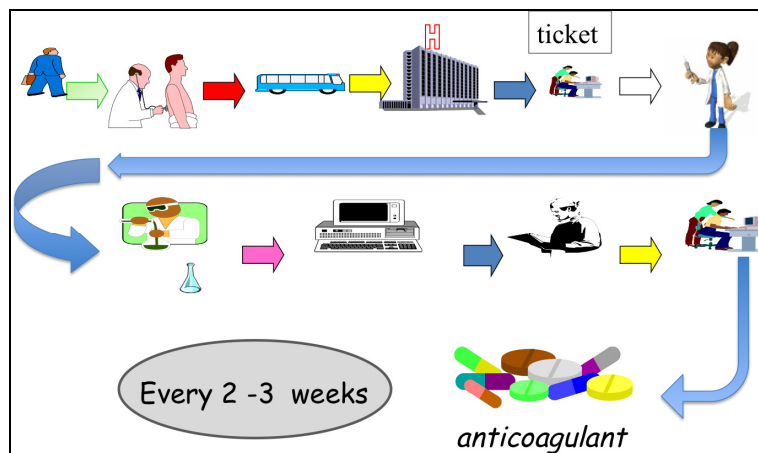


Fig. 5-16: Oral Anticoagulation Therapy workflow

The "trial & error" basis of the methods currently in use by physicians to fine tune the dosage for a given patient along with the response's variability due to genetic and behavioural factors, can result in out of range periods and therefore in a non negligible risk of thromboembolic and bleeding events. In particular, individual characteristics and behaviour, such as sex and diet, account for the variation in appropriate warfarin dose across individuals.

Indeed warfarin initiation is associated with one of the highest adverse events for anticoagulation drug due to high inter-individual variability. About 50% of patients fail to stabilize within the therapeutic range: for this reason most of these patients even with no contraindication to warfarin therapy are not receiving it because physicians are reluctant to initiate it in elderly patients or with risk of bleeding. This fact has strong negative clinical applications in that the absolute benefit of Oral Anticoagulation increases as patient get older (Rosove et al., 2009).

Recent research shows that genetic tests can, to some extent, identify which patients require higher and lower doses and may be a way to reduce bleeding or thrombotic events from warfarin. In particular, genotypes of patients have been recently suggested in order to understand their variability and control the dose-INR relationship, particularly in the induction phase. This fact has been also recognized by Food and Drug Administration (FDA) whose labelling for Warfarin 2007 reads: "It cannot be emphasized too strongly that the treatment of each patient is a highly individualized matter".

A notable contribution to patient genotyping is: (Schwarz et al., 2008) where it is shown that genetics variants of the enzyme that metabolized Warfarin cytochrome P450 CYP2C9 and VKORC1 contribute to differences in patients' response. Basically the same results have been obtained in a wide range genetics investigation (Sconce et al., 2005; Voora et al., 2005; Wadelius et al., 2007; Anderson 2007; Gage et al., 2008; Wadelius et al., 2009).

While there is a relative large agreement of the value of genotypes for the induction phase (Lesko, 2008), the debate is still open on its effectiveness in the long term therapeutic management (Garcia et al., 2009; Eckman et al., 2009).

### 5.3.2 Patient Profiling: Drug Sensitivity Index

To improve the characterization of the patient we introduce an index, called *Drug Sensitivity* ( $D_{sens}$ ), with the aim to capture the dose-INR relationships that better describes the patient behaviour. This index is represented by the ratio between dose and INRs variations, as follows:

$$D_{sens} = \frac{\sum INR_i}{\sum \Delta d_i} \quad (5-25)$$

Where:

$$\Delta d_i = \frac{(d_{i+1} - d_i)}{7} \quad \Delta INR_i = \frac{(INR_{i+1} - INR_i)}{Nd} \quad (5-26)$$

As INR measurements are not taken at regular intervals the dose values are replaced by their daily variations ( $\Delta d_i$ ), and the INR values by  $\Delta INR_i$  (computed with the above formula) where  $Nd$  is the number of days between consecutive measurements. Note that a negative value of  $D_{sens}$  means that patient is not responding to the therapy because increasing (decreasing) doses are likely to correspond to increasing (decreasing) INR values. In this case a high absolute value of  $D_{sens}$  correspond to patients whose response in highly unpredictable. Positive values of  $D_{sens}$  indicate that patient is responding to the therapy, in this case the absolute value indicate the response sensitivity with respect to the dosage, patients falling in this class have more predictable drug response



Given a sequence  $X$ , consisting of  $N$  instantaneous INR measurements:

$$X = INR_1, INR_2, \dots, INR_N \quad (5-27)$$

We must choose values for two input parameters,  $m$  and  $r$ , to compute the approximate entropy,  $ApEn(X_N, m, r)$ , of the sequence. In particular,  $m$  specifies the pattern length, and  $r$  defines the criterion of similarity.

Consequently, we construct a new series  $\vec{X}$  of vectors (or patterns)  $\overrightarrow{INR}_i$ :

$$\vec{X} = \overrightarrow{INR}_1, \overrightarrow{INR}_2, \dots, \overrightarrow{INR}_{N-m+1} \quad (5-28)$$

Where

$$\overrightarrow{INR}_i = (INR_i, INR_{i+1}, INR_{i+2}, \dots, INR_{i+m-1}) \quad (5-29)$$

Each vector  $\overrightarrow{INR}_i$  represents a subsequence of size  $m$  of INR measurements beginning at measurement  $i$ .

As a consequence, after the selection of the threshold distance  $r$ , we can define that two vector  $\overrightarrow{INR}_i$  and  $\overrightarrow{INR}_j$  are *similar* if the difference between any pair of corresponding measurements in the vectors is less than  $r$ , i.e., if

$$|INR_{i+k} - INR_{j+k}| < r \quad \forall k, 0 \leq k < m-1 \quad (5-30)$$

In this way, given the threshold distance  $r$ , the probability of a vector  $\overrightarrow{INR}_i$  of size  $m$  to be similar with a vector  $\overrightarrow{INR}_j$  of the same size is:

$$C_i^m(r) = \frac{\sum_{j=1}^{N-m+1} \Theta(i, j, m, r)}{N - m + 1} \quad (5-31)$$

where

$$\Theta(i, j, m, r) = \begin{cases} 1 & \text{if } |INR_{i+k} - INR_{j+k}| < r \\ 0 & \text{otherwise} \end{cases} \quad (5-32)$$

The quantity  $C_i^m(r)$  is the fraction of patterns of length  $m$  that look like the pattern of the same length that begins at interval  $i$ . We can calculate  $C_i^m(r)$  for each pattern in  $\vec{X}$ .

Finally, if we define

$$\Phi^m(r) = \frac{\sum_{i=1}^{N-m+1} \ln C_i^m(r)}{N - m + 1} \quad (5-33)$$

The approximate entropy, given  $m$  and  $r$ , is the difference:

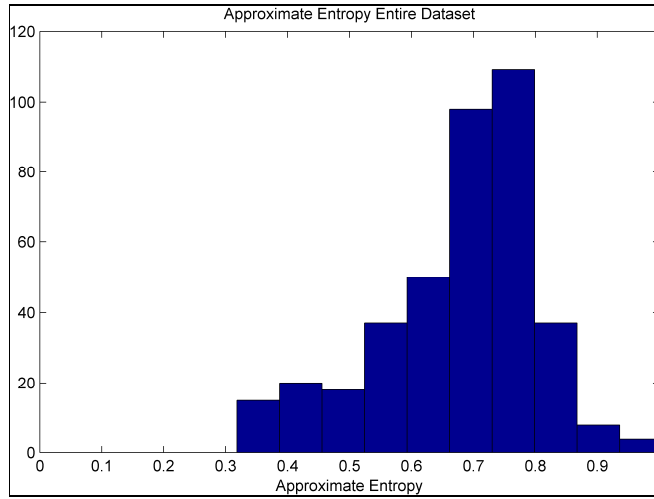
$$ApEn(m, r) = \Phi^m(r) - \Phi^{m+1}(r) \quad (5-34)$$

If the measurements' series is highly irregular, the occurrence of similar patterns will not be predictive for the following measurements and  $ApEn$  will be relatively large.

After different tests on our data and following the opinion of physician (Ho et al., 1997), we set the parameter  $m$  equals to 5 and  $r = 0.2 \cdot \sigma(X)$  where  $\sigma$  represents the standard deviation of patient's INR measurements.

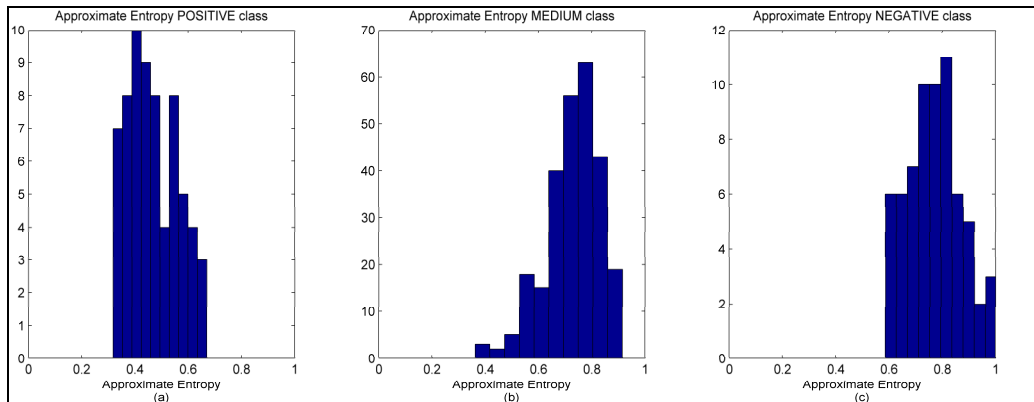
Results confirm the division in three classes already proposed before i.e. negative, medium and positive. The approximate entropy distribution for the entire dataset is illustrated in Fig. 5-18.





**Fig. 5-18:** Approximate Entropy distribution for entire dataset

As illustrated in Fig. 5-19, positive class (Fig. 5-19 (a)) has a low value of entropy (from 0.3 to 0.65) while negative class (Fig. 5-19 (c)) has high entropy values (from 0.6 to 1). This confirms the obtained classes and in particular the fact that patient in negative class has INR measurements with a lot of unpredictability fluctuations.



**Fig. 5-19:** Approximate entropy for each  $D_{sens}$  class

Approximate entropy has been computed also for better characterize patients with respect to different other features as for example age. Fig. 5-20 illustrates the approximate entropy distribution for patient older than 75 years and those younger than 75 years. In this case we can see that elderly patients have, like some literature works affirm (Rosove et al., 2009), a higher entropy than the young ones.

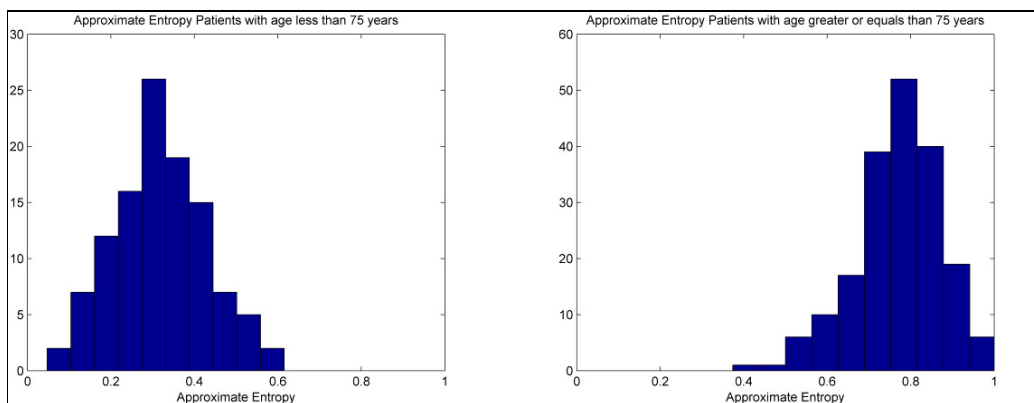


Fig. 5-20: Approximate Entropy for patients with different age

### 5.3.3 Traditional Clustering Approach:

#### a Modified Version of Mod-K-Prototype

After the definition of  $D_{sens}$  index that characterizes patients, a first analysis of available OAT data is made by a traditional clustering algorithm with the aim to find new interesting properties into data.

In particular, this case study is an example of a typical problem for biomedical data analysis explained in section 2.2: the need of clustering for mixed data types. In this case, in fact, patients' data are both categorical (like gender, concomitant therapy, etc.) and numerical (like doses, INR values, age, etc.).

We use a specialized version of the algorithm "modified k-prototypes" (Bushel et al., 2007) for handling mixed data, that we will call "OAT- Mod-k-prototypes".

Like already described the approach is based on the construction of an objective function obtained from the sum of the squared Euclidean distance for numeric data and simple matching distance for categorical values in order to measure dissimilarity of the instances (patients). Separate weighting terms are used to control the influence of each data type.

A cluster's prototype (centroid) is formed from the mean of the values for numeric features and the mode of the categorical values of all the samples in the group.

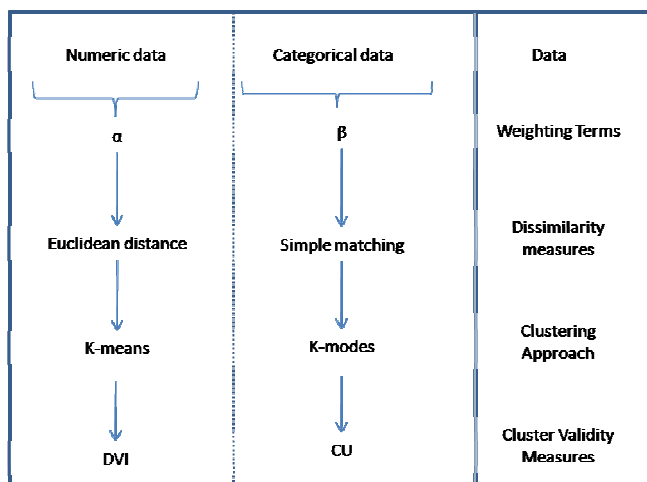


Fig. 5-21: schema of the OAT Mod-k-prototypes algorithm

In Fig. 5-21 are represented the components of the modified k-prototypes algorithm for our OAT mixed dataset.

The mod-k-prototypes objective function is formulated as:

$$d = \alpha \sum_{j=1}^{m_n} (p_{ij}^n - q_{ij}^n)^2 + \beta \sum_{j=1}^{m_c} (p_{ij}^c - q_{ij}^c) \quad (5-35)$$

where  $p_i$  is the  $i^{\text{th}}$  patient ( $i = 1$  to  $N$  number of patients),  $q_l$  is the  $l^{\text{th}}$  centroid, ( $l=1$  to  $k$  number of clusters),  $m_n$  is the number of numeric attributes and  $m_c$  is the number of categorical attributes;  $\alpha$  and  $\beta$  denote the weights ( $W$ ) for the numeric and categorical data domain dissimilarity measures, respectively.

The weights for data type  $t$  ( $t=\text{numerical, categorical}$ ) at the  $n^{\text{th}}$  step are called  $W_t[n]$  where if  $t$  is referred to numerical attributes represents  $\alpha$  while if it is referred to categorical values represents  $\beta$ .

These weights are adapted and modified respect to equation (2-7) as follows:

$$W_t[n] = \begin{cases} \frac{1}{2} & n = 0 \\ (1-\tau) \cdot W_t[n-1] + \tau \cdot \text{avecorr}(p^d, q^d) & \text{otherwise} \end{cases} \quad (5-36)$$

where  $\tau$  is the exponential weighting update factor in the range  $[0,1]$  and  $\text{avecorr}(p^d, q^d)$  is the average correlation coefficient between the instances and the centroids based on the feature values from type  $t$ , as defined in equation (2-8).

In our setting  $\tau=0.13$  as shown in Fig. 5-22.

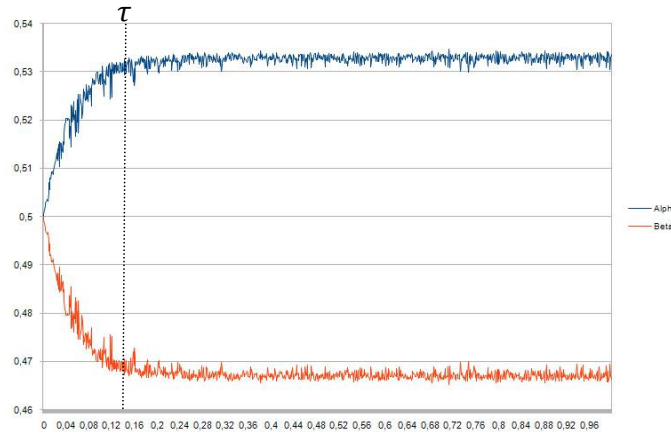


Fig. 5-22:  $\alpha$  and  $\beta$  parameter

Finally, the dynamic validity index and category utility index (DVI\_CU), defined in subsection 1.4.1.2 is used to both determine the optimal number of clusters and for the obtained clusters.

### 5.3.3.1 OAT-Mod-K-Prototypes Results

In order to investigate better the importance of genomic features in the characterization of the patient, we tested this algorithm by considering data with or without genetic features which represent the polymorphism of gene CYP2C9 and VKORC1. Feature CYP2C9 can assume values [WT, AC, CT] and VKORC1 the

values [WT, CT, TT]. We call the configuration without genomic data  $\Omega_1$  and the other  $\Omega_2$ .

$\Omega_1$  and  $\Omega_2$  are clustered using the OAT-modified-k-prototypes algorithm with values of  $k = 2, \dots, N$ .

As shown in Fig. 5-23 we have the minimum value of DVI\_CU index, for  $\Omega_1$ , with  $k=3$  and  $DVI\_CU = 1.12$ .

The results shown in Fig. 5-24 are those obtained on the dataset  $\Omega_2$ : in this case minimum DVI\_CU index is obtained with  $k=3$  and  $DVI\_CU=1.08$ .

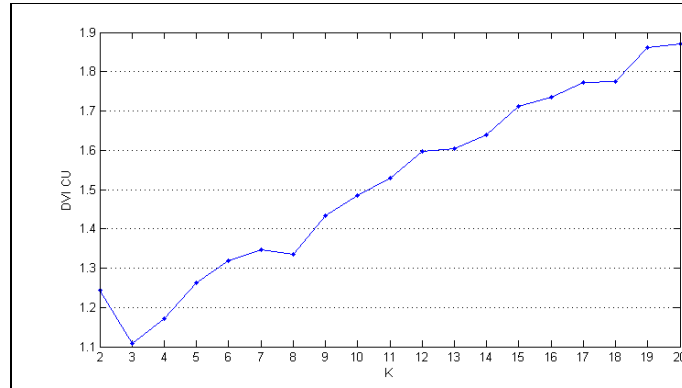


Fig. 5-23: DVI\_CU index variation for k from 0 to 20 for dataset  $\Omega_1$

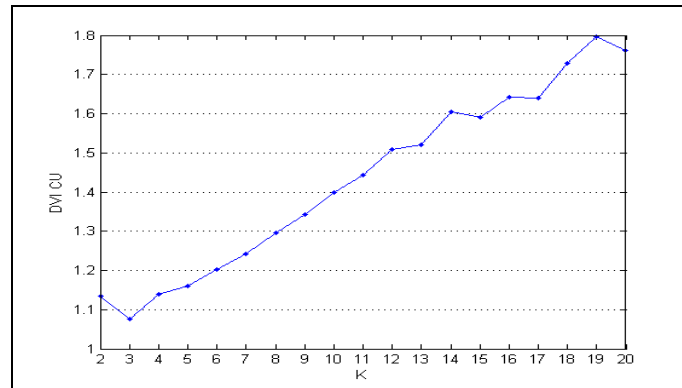


Fig. 5-24: DVI\_CU index variation for k from 0 to 20 for genomic dataset  $\Omega_2$

The obtained number of clusters (3) has confirmed our initial subdivision of patients in three principal categories (negative, medium and positive).

To evaluate clustering results we use two of the traditional classification-oriented evaluation measures, already defined in subsection 1.4.1.1: *F-Measure* which combines the Precision and Recall and *Entropy*.

The used set of labels  $L$  is based on the three  $D_{sens}$  classes defined before.

		<i>F-measure</i>	<i>Entropy</i>
OAT-modify-k-prototypes	$\Omega_1$	0.521	1.48
	$\Omega_2$	<b>0.563</b>	<b>1.35</b>

Tab. 5-8: F-measure and entropy results for OAT modify k prototypes algorithm

The results obtained by OAT-mod-k-prototypes are summarized in Tab. 5-8, where we can see that better results, in terms of both f-measure and entropy, are those achieved by genomic configuration  $\Omega_2$ .

### 5.3.4 The Proposed Relational Clustering Framework

The clustering approach presented in the previous subsection belong to the category of distance-based approaches, in which the objective function is based only on the minimization of a combination of two distance measures, one for categorical data and the other for the numerical ones.

Also in this case study we instantiate the proposed relational clustering framework (presented in subsection 4.4), considering behavioural relationships between patients as background knowledge that can be included into the clustering objective function. A representation of the proposed framework for this clinical application is illustrated in Fig. 5-25.

The main aim of this application is based on the creation of groups of patients based on their clinical features, by constraining the clustering process in order to consider also the behavioural relationships between couples of patients. This is motivated by the fact that we want find not only clusters of patients clinically similar, but also cluster of patients sharing a similar behaviour, as, for example, similar response to oral anticoagulation therapy.

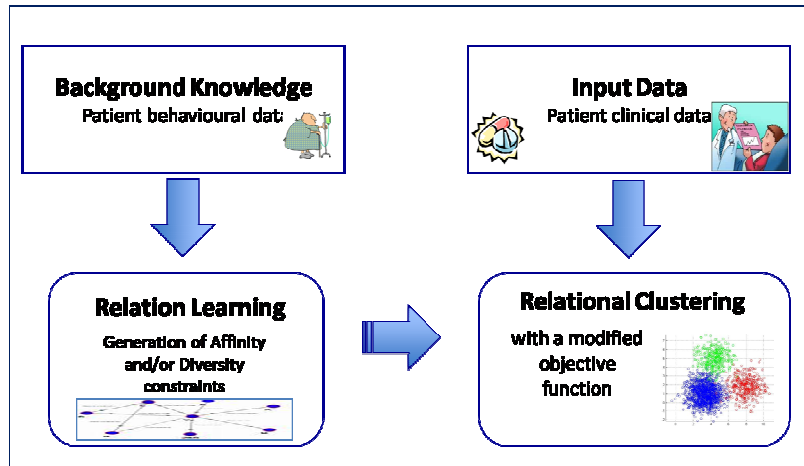


Fig. 5-25: Instantiation of general relational clustering framework

The computational process of the proposed framework is illustrates in Fig. 5-26.

The OAT dataset can be viewed as a set into the space  $R^{m+n}$  where  $m$  is the behavioural feature number and  $n$  the clinical features number.

$$\Omega = \{p | p = (p^{Bf}, p^{Cf}), p^{Bf} \in R^m, p^{Cf} \in R^n\} \quad (5-37)$$

where  $p$  represents a patient,  $p^{Bf}$  corresponds to the behavioural features value into a space  $R^m$  and  $p^{Cf}$  denotes the clinical features value into the space  $R^n$ .

We can therefore define  $\Omega^{Bf}$  and  $\Omega^{Cf}$  as the set of the patients represented through their behavioural and clinical features respectively:

$$\Omega^{Bf} = \{p^{Bf} | p = (p^{Bf}, p^{Cf}), p \in \Omega\} \quad (5-38)$$

$$\Omega^{Cf} = \{p^{Cf} | p = (p^{Bf}, p^{Cf}), p \in \Omega\} \quad (5-39)$$

In particular in the behavioural features set we include numerical data as: average and variance of INR measurements, average and mean of doses.

On the other side, the set of clinical features is composed only by categorical data i.e. age, gender, evidence leading to OAT, concurrent medications.

In this case study since we have both numerical and categorical data we use two different distance measures. In particular, for numerical data we use the traditional Euclidean distance, described into subsection 1.1. On the other side, for categorical data we use the simple matching distance measure, already used in the modified-k-prototypes algorithm.

### 5.3.4.1 Relations Learning

A relation among two patients can be an *Affinity* or a *Diversity* link. In this case study with *Affinity link* we want to explain the relation between two patients that shared similar behavioural profile and with *Diversity link* we want to underline the fact that the two patients don't share a common behaviour.

Given that we don't know any behavioural relationships between patients in advance, we can learn *Affinity* and *Diversity* link weight by formulating a clustering problem that can be solved by the K-Means algorithm.

Given the elements  $p_i^{Bf}$ , a set of clusters  $C_j$  with  $j = 1 : J$ , the clustering problem can be defined as in equation (1-11), where we must allocate each element  $p_i^{Bf}$  in a cluster  $C_j^{Bf}$  such that the intra-cluster distance is minimized and the inter-cluster distance is maximized.

The resulting clusters of patients permit us to build two relations matrices, called  $R^A$  and  $R^D$  where:

- $R^A$ : is a matrix whose elements  $r_{ik}^A$  represent the weights of the *Affinity links* between patients belonging to the same cluster and will suggest in the following phase that two patients should be placed in the same module, because they share some behavioural features. The element  $r_{ik}^A$  is given by:

$$r_{ik}^A = \begin{cases} d(p_i^{Bf}, p_k^{Bf}) & \text{if } p_i^{Bf} \wedge p_k^{Bf} \in C_\alpha^{Bf} \\ 0 & \text{otherwise} \end{cases} \quad (5-40)$$

- $R^D$ : each element  $r_{ik}^D$  of this matrix represents the weights of the *Diversity links* between patients belonging to different clusters and will suggest that in the following phase two patients should not be placed in the same module because they have different behaviour (for instance they respond in a different way to warfarin therapy).

More formally,  $r_{ik}^D$  is defined as the distance (computed as in (5-10)) between the behavioural profile of patient  $i$  and that of patient  $k$ , i.e.

$$r_{ik}^D = \begin{cases} d(p_i^{Bf}, p_k^{Bf}) & \text{if } p_i^{Bf} \in C_\alpha^{Bf} \text{ and } p_k^{Bf} \in C_\beta^{Bf} \neq C_\alpha^{Bf} \\ 0 & \text{otherwise} \end{cases} \quad (5-41)$$

These two matrices will be used in the subsequent objective function optimization over the clinical features space.

### 5.3.4.2 Relational Clustering

The second component of relational clustering process, illustrated in Fig. 5-26, is based on grouping patients on the basis of their clinical features taking into account the relations, based on patients behavioural features, learned in the previous step.

In particular, we want to minimize the sum of the distances between instances, expressed by patients' clinical features, penalized by a function that takes into account *Affinity* and *Diversity* links.

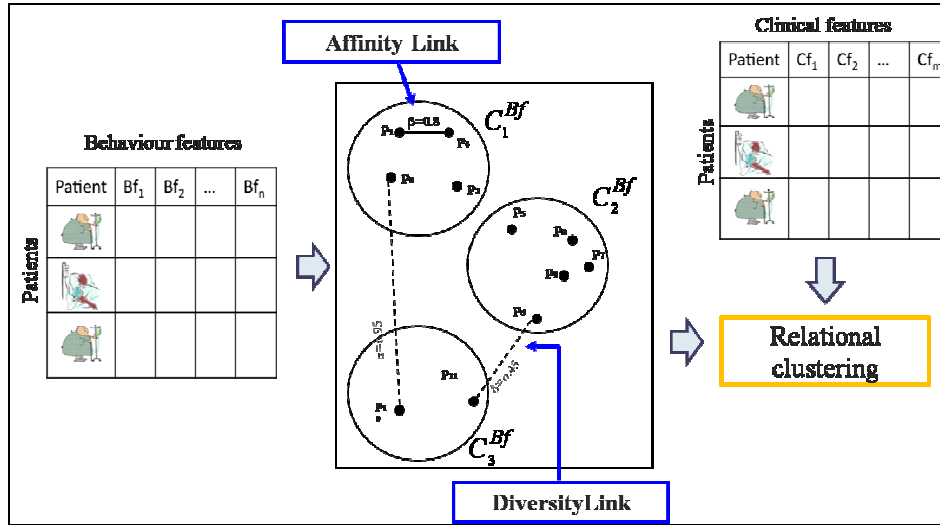


Fig. 5-26: Computational process of the proposed relational clustering framework

Let  $p_i^{cf}$  be a given patients represented by its clinical features, like age and therapy and  $C_j^{cf}$ , with  $j = 1 : J$ , be a set of clusters.

For solving the clustering problem we must define the assignment matrix, called  $Z$  (as in equation ( 1-10)), where each element  $z_{ij}$  is equal to:

$$z_{ij} = \begin{cases} 1 & \text{if } p_i^{cf} \in C_j^{cf} \\ 0 & \text{otherwise} \end{cases} \quad (5-42)$$

The problem can be formulated as follows:

$$\min \sum_{j=1}^J \left[ \sum_{ik=1}^{\Omega} \left[ d(p_i^{cf}, p_k^{cf}) z_{ij} z_{kj} - d(p_i^{cf}, p_k^{cf}) z_{ij} (1 - z_{kj}) + z_{ij} z_{kj} r_{ik}^D d(p_i^{cf}, p_k^{cf}) + z_{ij} (1 - z_{kj}) r_{ik}^A d(p_i^{cf}, p_k^{cf}) \right] \right] \quad (5-43)$$

The optimization problem can be solved through an adaptation of K-means heuristic reported in the starting chapter.

In this way if an *Affinity link* (or a *Diversity link*) is not preserved, the objective function is penalized according to their respective weight ( $r_{ik}^A$  or  $r_{ik}^D$ ) and the distance between patients  $i$  and  $k$  represented in terms of their clinical features,  $d(p_i^{cf}, p_k^{cf})$ .

### 5.3.4.3 Obtained Results

In the analysis of relational clustering results we use the two dataset configurations  $\Omega_1$  and  $\Omega_2$  based on the inclusion or exclusion of genomic features as already used in the results analysis of OAT-modify-k-prototypes algorithm. Also in this case we evaluate the resulting clusters using two traditional classification oriented measures: *F-Measure* and entropy (defined in subsection 1.4.1.1) based on the set of class label  $L$  constituted by the three drug sensitivity class, since, as already investigate, this index characterize the patient behaviour. In the table below we have reported results that we have already seen in the previous subsection and Relational clustering results.

		<i>F-measure</i>	<i>Entropy</i>
OAT-modify-k-prototypes	$\Omega_1$	0.521	1.48
	$\Omega_2$	0.563	1.35
Relational Clustering	$\Omega_1$	0.600	1.25
	$\Omega_2$	<b>0.630</b>	<b>1.01</b>

**Tab. 5-9:** F-measure and entropy results for OAT modify k prototypes algorithm

As we can see relational clustering approach obtain more purely clusters in respect to the traditional one. Maybe the increments obtained are due to the inclusion of underlined behavioural relationships between patients. In fact, using traditional approach we could obtain clusters of patients with the same age or therapy or other clinical features, but these patients can differ greatly each other under the point of view of their therapeutical behavioural. Using behavioural and not-behavioural links, patients with same behavioural features are forced to be in the same cluster.

Another interesting characteristic deducible from results is that genomic features increment clustering performance both in term of F-measure and Entropy.

### 5.3.5 Further Analysis on the Data Set

After an analysis of OAT data using unsupervised machine learning techniques we have further analyzed it by using supervised techniques based on drug sensitivity classes.

Analyzing literature, for the best of our knowledge, we haven't found any application of unsupervised techniques on oral anticoagulation problem. On the other side, different patient classification models, based on personal and clinical data, have been proposed in (Carney et al., 2005; Mc Donald et al., 2008). However, these traditional machine learning applications classifies patients on their average INR value (below, in and over patient range) without considering their drug sensitivity.

In this Thesis we investigate classification models using drug sensitivity index, explained above, as class variable.

In order to build a classification model we considered the following features: personal data (age and gender), OAT therapeutic data (drug used for OAT therapy and medical evidence leading to OAT) and concomitant medication.

We train and test, using 10-fold cross validation, four different machine learning classification algorithms (Multi Layer Perceptron (MLP), Support Vector



Machines (SVM), K-Nearest Neighbourhoods (kNN) and Bayesian Networks (BN)). In this study, a particular configuration for MLP is used: it had two hidden layers and a high momentum value than usual to try to alleviate the potential problem of reaching only a local optimal solution, rather than a global optimum. Therefore SVM, a machine learning approaches based on multiple regression, was configured using a non-linear kernel function.

A particular version of KNN is used: the neighbours (3 in our configuration) were weighted by the inverse of their distance. Finally, for finding the conditional probability tables of the Bayesian Network we use a Simple Estimator and for finding a well scoring BN structure we use a Genetic Search that works by having a population of Bayesian Network structures and allow them to mutate and apply cross over to get offspring.

For our experiments we use the Weka ([www.cs.waikato.ac.nz/ml/weka](http://www.cs.waikato.ac.nz/ml/weka)) implementation of the used classification algorithms.

### 5.3.5.1 Classification Results

In this paragraph we will present the obtained classification results in term of correctly classified instances (CCI) and F-measure (the weighted harmonic mean of precision and recall), two extensively used metrics in supervised learning. The first step that we have done to better compare our classification results with literature is based on building classification models based only on INR class (low, in or over range). We report the obtained results in the table below.

	<i>INR</i> based Classification			
	MLP	SVM	kNN	BN
% CCI	59,30%	59,30%	58,47%	58,79%
F-measure	0,5513	0,5818	0,5521	0,5380

**Tab. 5-10:** INR based classification results

The next phase of our supervised analysis is based on the training and testing of classification models based on drug sensitivity class label. Obtained results are presented in Tab. 5-11.

	<i>DRUG SENSITIVITY</i> based Classification			
	MLP	SVM	kNN	BN
% CCI	60.61%	64.06%	59.32%	62.29%
F-measure	0.581	0.595	0.578	0.589

**Tab. 5-11:** Drug sensitivity based classification results

As we can observe in the table, there is an increment in terms of both F-measure and CCI for drug sensitivity based classification respect to the traditional one proposed in Tab. 5-10 .

To characterize better the behaviour of a patient we compute INR average and variance of a time course of 6 INR measurements and include both these data in the feature set. So, we built new classification models with this new feature set

and the obtained results, reported in Tab. 5-12, are better both in term of CCI and F-measure.

	<b>DRUG SENSITIVITY</b> based Classification			
	MLP	SVM	kNN	BN
% CCI	63.71%	68.70%	64.30%	64.46%
F-measure	0.6214	0.613	0.598	0.611

**Tab. 5-12:** Drug sensitivity based classification results with new features

The model thus learned i.e. with the full set of features has been applied also in the induction phase i.e. without considering INR values. Obtained results are reported in Tab. 5-13.

Comparing these results with those reported in Tab. 5-11, we can see that models learned using the two additional features about INR are better in term of CCI and f-measure than those learned without these two features.

	<b>DRUG SENSITIVITY</b> based Classification			
	MLP	SVM	kNN	BN
% CCI	61.61%	65.26%	60.88%	63.8%
F-measure	0.5964	0.61	0.584	0.597

**Tab. 5-13:** Induction phase: Drug sensitivity based classification results

Also in this case, like in clustering analysis, another investigation has been made taking into account genomic data.

We now present classification results obtained including the two genomic feature regarding polymorphism of CYP2C9 and VKORC1 genes in the feature set previously introduced. This dataset configuration is the same that we called, in the previous section, as  $\Omega_2$ .

Also with this dataset four different tests are performed.

The first one is focalized on INR based classification. Obtained results are reported in the table below.

	<b>INR</b> based Classification			
	MLP	SVM	kNN	BN
% CCI	60,12%	61,95%	60,24%	59,44%
F-measure	0,573	0,601	0,574	0,579

**Tab. 5-14:** INR based classification results on  $\Omega_2$  dataset configuration

As in the tests presented previously, in the induction phase first stage we do not use INR average and variance. Results obtained at this step are reported in Tab. 5-15.

	<b>DRUG SENSITIVITY</b> based Classification			
	MLP	SVM	kNN	BN
% CCI	61.66%	65.6%	62.3%	63.5%
F-measure	0.591	0.62	0.58	0.60

**Tab. 5-15:**  $D_{sens}$  based classification with genomic data results.  
In this phase INR average and variance are not considered.

Results obtained using the complete set of features (including INR average and variance) are reported in Tab. 5-16. A new improvement is visible compared to the results in Tab. 5-12 .

	<b>DRUG SENSITIVITY</b> based Classification			
	MLP	SVM	kNN	BN
% CCI	68.61%	74.41%	79.07%	75.58%
F-measure	0.675	0.747	0.665	0.645

**Tab. 5-16:**  $D_{sens}$  based classification with complete genomic data results

Also in this case, the model thus learned i.e. with the full set of features has been applied also in the induction phase i.e. without considering INR values. Obtained results are reported in the table below.

	<b>DRUG SENSITIVITY</b> based Classification			
	MLP	SVM	kNN	BN
% CCI	62.2%	66.8%	63.1%	64.1%
F-measure	0.62	0.64	0.60	0.625

**Tab. 5-17:**  $D_{sens}$  based classification with genomic data results.  
In this phase INR average and variance are not considered.

### 5.3.5.2 Drug Sensitivity Classes Characterization Based on Genetic Features

Analyzing classification results presented in the last paragraph, we can understand that genomic data allow a better characterization of patient's behaviour. This confirms both the first data analysis done with the entropy measure and the results obtaining thru the relational clustering framework.

In Tab. 5-18 we provide the distribution of genomic variants into the three drug sensitivity classes described before.

An interesting feature is that a lot of patients with a polymorphism on both genes belong to negative drug sensitivity class. Therefore wild type patients are predominantly in positive and medium drug sensitivity classes and patients with only one polymorphism are distributed principally in medium.

	Genes		Total Patient Number	Drug Sensitivity class		
	CYP2C9	VKORC1		POSITIVE	MEDIUM	NEGATIVE
<b>WILD TYPE</b>	WT	WT	65	51,72%	44,83%	3,45%
<b>ONE POLYMORPHISM</b>	WT	TT	53	34,21%	50,00%	15,79%
	WT	CT	96	34,78%	60,87%	4,35%
	CT	WT	31	8,33%	75,00%	16,67%
	AC	WT	6	0,00%	25,00%	75,00%
<b>TWO POLYMORPHISMS</b>	CT	CT	32	10,00%	30,00%	60,00%
	CT	TT	16	0,00%	83,33%	16,67%
	AC	CT	18	14,29%	14,29%	71,43%
	AC	TT	8	0,00%	16,67%	83,33%

Tab. 5-18: Genomic variant distribution in the three  $D_{sens}$  classes

We studied also patients belonging to each class and we report an example of INR measurements of one patients of each class.

In this way we want to underline not only the negative behaviour of a patient, but also the unpredictability of his INR values. In Fig. 5-27 are plotted INR values of a wild type patient belonging to the positive drug sensitivity class. Comparing this plot with that reported in Fig. 5-29 is possible to see that the hemorrhagic or thrombotic risk of a patient in negative  $D_{sens}$  class is higher than that of a positive  $D_{sens}$  patient.

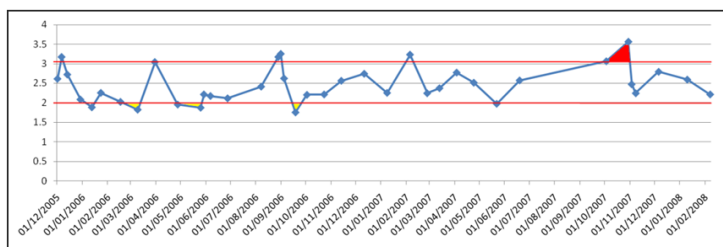


Fig. 5-27: Wild type patient, positive Drug Sensitivity class

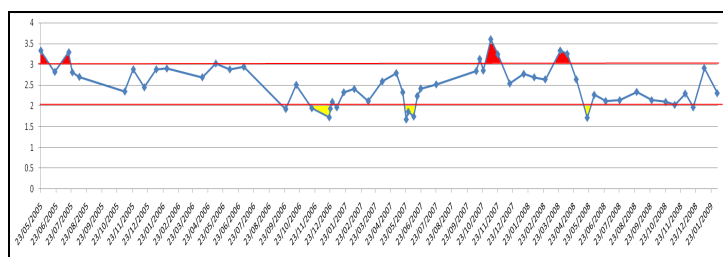
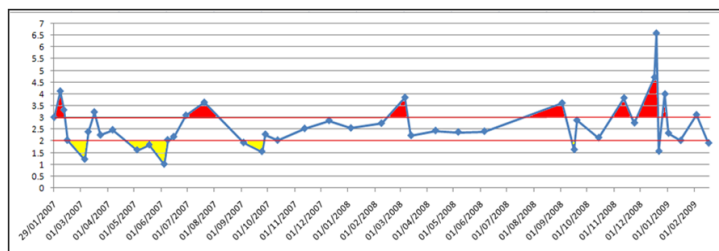


Fig. 5-28: Patient with two polymorphisms (gene CYP2C9: AC; gene VKORC1: TT), medium Drug Sensitivity class



**Fig. 5-29:** Patient with two polymorphisms (gene CYP2C9: CC; gene VKORC1: CT), negative Drug Sensitivity class

In conclusion, in this subsection we have applied different machine learning techniques on the clinical problem of oral anticoagulation therapy.

In particular, after the introduction of Drug Sensitivity index (used to better characterize patients), we instantiated the relational clustering framework for finding groups of patients with similar behavioural and clinical features.

Results are compared with those of a traditional distance based approach (OAT-mod-k-prototypes): improvements are obtained in terms of F-measure and entropy.

In a second phase, classification models based on drug sensitivity classes have been trained and tested.

Analyzing both clustering and classification results we understand the importance of genetic information, that increments the performance of the algorithms and therefore help us to better characterize patients.

## Conclusion

In this Thesis, we have addressed the problem of incorporating into traditional clustering algorithms available background knowledge, gathered from difference sources.

We have investigated three main methodological challenges in clustering namely: feature selection, distance measure using mixed data types and knowledge integration into the clustering procedure.

The main methodological achievements of this work can be synthesized as follow:

- Concerning the goal of feature selection, a new genetic programming based framework has been proposed to automatically identify and remove irrelevant and redundant information. Interesting results has been obtained on different life science case studies.
- With respect to the second challenge, a modified clustering version for mixed data type has been proposed. The traditional clustering objective function has been modified to take into account a weighted combination of numerical and categorical distances, with the aim to control the influence of each data domain on the clustering of the instances.
- As far as the integration of background knowledge in the clustering process is concerned, a relational clustering framework has been proposed, characterized by two main components: “relation learning”, in which relations are discovered and learned from domain background information and “relational clustering” in which the learned relationships are included into the clustering process modifying the clustering objective function.

The proposed methodologies have been validated on three different applications in the life science domain: the first focused at the identification of modules of genes with similar regulatory profiles; the second one is a pharmacogenomics application with the aim of defining a model which, given the gene expression profile related to a specific tumors tissue, could help in selecting a set of most responsible drugs; finally the proposed relational framework is applied to a clinical application for grouping patients undergoing the Oral Anticoagulation Therapy in order to profile them based on their behavior and clinical features.

The experiments done on all the case studies mentioned above demonstrate that using background knowledge the clustering accuracy improves and that the application of the relational clustering algorithm yields better results than traditional approaches.

## Research Perspectives

There are different interesting research topics which stem from the ideas presented in this Thesis.

An interesting research direction could be the extension of the proposed relational clustering framework for clustering data with respect to more dimensions, like biclustering and 3-clustering.

A further interesting development could be the extension of the relational clustering framework to include background information with richer structure, like the approach proposed in (Taskbar et al., 2001), who developed a relational approach based on Probabilistic Relational Models (PRM) introduced by (Friedman et al., 1999). In a simple way, PRM can be viewed as a Bayesian Network extended to the relational domain. In this way an additional development will be aimed at defining new similarity measures between objects that consider the relational structure.

There are a number of other interesting future directions regarding the case studies presented in this thesis; below we present the possible ones for each case study:

#### *Learning Transcriptional Regulatory modules*

In the system biology domain a lot of different information are available. We use regulatory information to find modules of genes that are co-regulated and co-expressed. Possible interesting background information that can be included into the clustering procedure is that became from the Gene Ontology (The Gene Ontology Consortium, 2000) database with functional gene annotations.

In this way we think that it is possible to increment the clustering performance and gene annotations will make easier the interpretation of clustering results.

#### *Detective the most effective cancer drug: NCI-60 dataset*

In this case study future works will be focused principally on a modification of the proposed relational clustering framework instantiation. In particular we would like to create an iterative procedure, that iteration after iteration select genes that most explain anticancer therapy responses. For this aim we can take as example the iterative procedure used in the first case study.

Other future contributions will concern in Vitro testing for validating drug prediction based on the relational clustering modules found by our framework.

Finally an interesting topic is the refinement of Bayesian Networks for predicting drug responses of those compounds that are in clinical use, after a strict collaboration with physicians.

#### *Oral Anticoagulation Therapy*

Using the relational framework in this application we are able to profile patients into three different sensitivity classes: negative, medium and positive.

This profiling will be useful for building “ad hoc” dosage algorithm. In particular, the workflow that we have thought is based on the following steps: the patient arrives at hospital, an INR measurement is done by the physician and, using all the information available, a drug sensitivity class is assigned to him. In particular, patient  $p$  could be associated to a specific class by using the minimum distance between  $p$  and the representative element of each class.

Different literature works like (Anderson et al., 2007; Gage et al., 2008), proposed pharmacogenomics dosing algorithms. We would suggest a personalized dosing algorithm based on patient’s class.

Thanks to patient profiling we will be also able to build a dosage system to suggest physician the correct drug dosage during the entire therapy. In this case we think that this problem can be viewed as a sequential optimization problem

and that the application of Markov Decision Processes (MDP) provides an appropriate model, since they take into account the long-term effects of each dosing action and the INR expected value.



## Appendix A: Data Resources

In this appendix we report a description of all data resources used in the case studies presented in this Thesis.

### ***A.1 Transcription Factors Data***

The complexity of the yeast cell's system for detecting and responding to environmental variation is only beginning to emerge and makes it a suitable model for analysis of more complex biological systems. Genes whose transcription is responsive to a variety of stresses have been implicated in a general yeast response to stress (Mager and De Kruijff, 1995; Ruis and Schuller, 1995). Other gene expression responses appear to be specific to particular environmental conditions. Several regulatory systems have been implicated in modulating these responses, but the complete network of regulators and the details of their actions, including the signals that activate them and the downstream targets they regulate, remain to be elucidated.

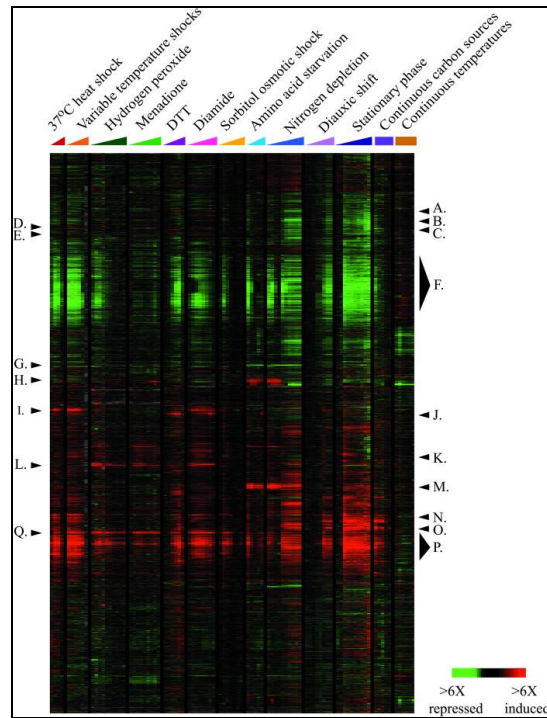
#### **A.1.1 Gasch et al., 2000 Dataset**

Gasch et al. 2000 used data came from DNA microarrays to analyze changes in transcript abundance in yeast cells responding to a panel of diverse environmental stresses.

In particular, they describe the global expression programs in response to a diverse set of stresses, including their specific features and a common response to all of the stressful conditions, termed the "environmental stress response" (ESR). These ESR are: heat shock, hydrogen peroxide, superoxide generated by menadione, a sulfhydryl oxidizing agent (diamide), and a disulfide reducing agent (dithiothreitol), hyper-osmotic shock, amino acid starvation, nitrogen source depletion, and progression into stationary phase. The severity of each condition was calibrated to preserve more than 80% cell viability, so that we could observe the expression programs in viable cells adapting successfully to a changing environment.

In their experiments, 173 different mRNA samples were analyzed by whole-genome microarray hybridization. Each microarray used in this study contains 6172 yeast genes.

The resulting table was organized by hierarchical clustering and displayed as in the figure below taken from (Eisen *et al.*, 1998). Briefly, their clustering algorithm arranges genes according to their similarity in expression profiles across all of the array experiments, such that genes with similar expression patterns are clustered together.



**Fig. A - 1:** (Gasch et al., 2000) dataset representation

In Fig. A - 1 each row of coloured boxes represents the variation in transcript abundance (expression) for each gene, and each column represents the variation in transcript (expression) levels of every gene in a given mRNA sample, as detected on one array. The variations in transcript abundance for each gene are depicted by means of a colour scale, in which shades of red represent increases and shades of green represent decreases in mRNA levels, relative to the unstressed culture, and the saturation of the colour corresponds to the magnitude of the differences. A black colour indicates an undetectable change in transcript level, and a gray colour represents missing data. A dendrogram constructed during the clustering process depicts the relationships between genes: the branch lengths represent the degree of similarity between genes based on their expression profiles. Genes that display similar patterns of gene expression over multiple experiments are thus grouped together on a common branch of the dendrogram and can also be recognized by an obvious pattern of contiguous patches of colour in the cluster diagram.

### A.1.2 Spellman et al., 1998 Dataset

In 1981 Hereford and co-workers discovered that yeast mRNAs oscillate in abundance during the cell division cycle (Hereford et al., 1981). To date 104 messages that are cell cycle regulated have been identified using traditional methods, and it was estimated that some 250 cell cycle-regulated genes might exist (Price et al., 1991). There are several reasons why genes might be regulated with the cell cycle. Such regulation might be required for the proper functioning of mechanisms that maintain order during cell division. Alternatively, regulation of these genes could simply allow conservation of resources.

Much of the literature has focused on the post-transcriptional mechanisms that control the basic timing of the cell cycle. However, there is also clear evidence that trans-acting factors play a critical role in the regulation of the abundance of many cell cycle-regulated transcripts.

In particular, (Spellman et al. 1998) used DNA microarrays and samples from yeast cultures synchronized by three independent methods: alpha factor arrest, elutriation, and arrest of a *cdc15* and *cdc28* temperature-sensitive mutant.

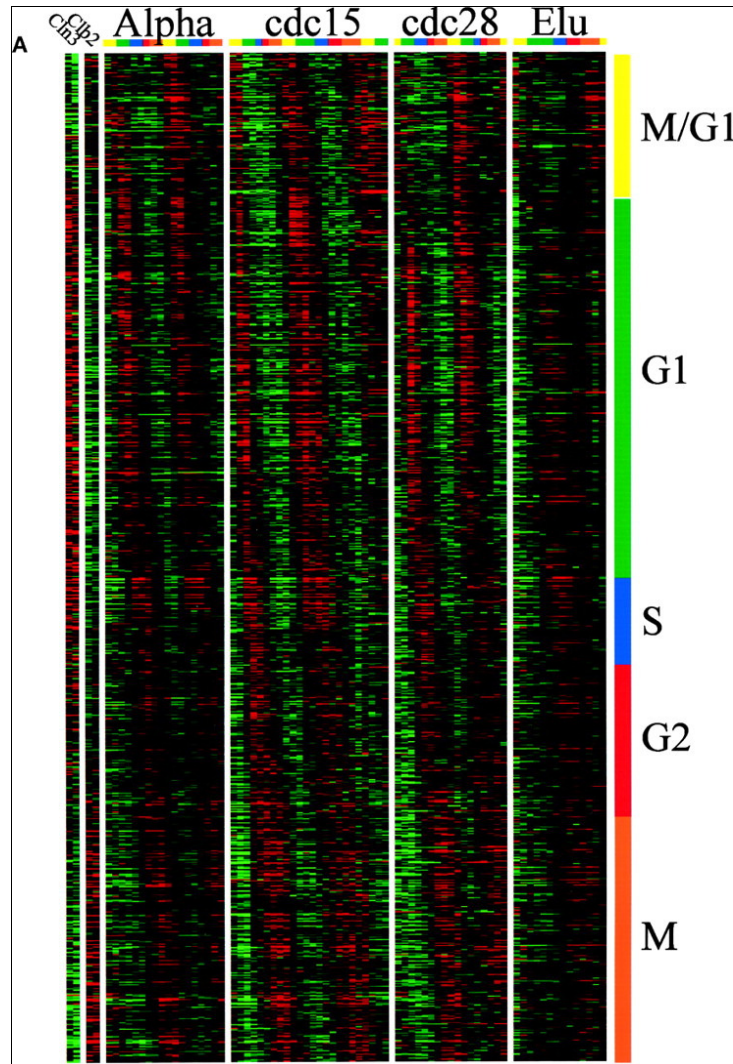


Fig. A - 2: (Spellman et al., 1998) data set representation

In Fig. A - 2 is represented the dataset composed by gene expressions during the yeast cell cycle. Genes correspond to the rows, and experiments are the columns. The ratio of induction/repression is shown for each gene such that the magnitude is indicated by the intensity of the colours displayed. If the colour is black, then the ratio of control to experimental cDNA is equal to 1, whereas the brightest colours (red and green) represent a ratio of 2.8-1. Ratios >2.8 are displayed as the brightest colour. In all cases red indicates an increase in mRNA abundance, whereas green indicates a decrease in abundance. Gray areas (when visible) indicate absent data or data of low quality. Colour bars on the right

indicate the phase group to which a gene belongs (M/G1, yellow; G1, green; S, purple; G2, red; M, orange). These same colours indicate cell cycle phase along the top. (A) Gene expression patterns for cell cycle-regulated genes.

In conclusion the microarray data used in this study are composed by the expression level of 6178 genes across 77 different experimental conditions. In particular, only 774 genes are cell cycle regulated which constitutes >10% of all protein-coding genes in the genome.

## A.2 NCI-60 Data

The NCI-60 data, commissioned by the National Cancer Institute U.S.A. and publicly available online (Scherf et al., 2000), consists of 60 cell lines from 9 kinds of cancers, all extracted from human patients.

The tumours considered in this panel derive from colorectal (8), renal (8), ovarian (6), breast (8), prostate (2), lung (8) and central nervous system (6) as well as leukaemia (6) and melanomas (8) cancer tissues.

The dataset is composed by two matrices: the Activity matrix (or simply *A-matrix*), that contains responses to pharmacologic treatments, and the Target matrix (or *T-matrix*) that contains the gene expressions data.

A representation of the dataset is visible in Fig. A - 3.

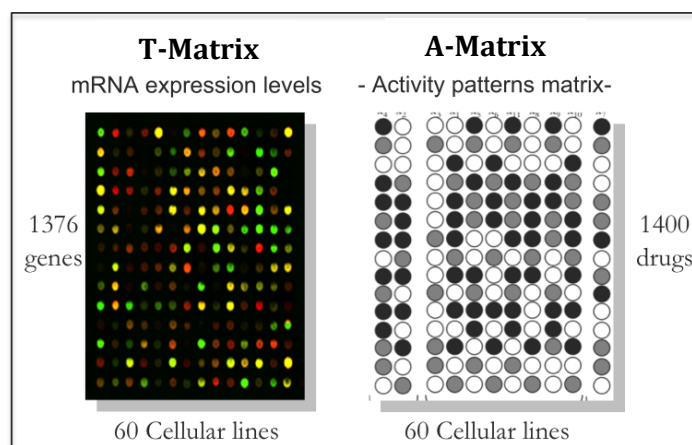


Fig. A - 3: Simplified schematic overview of NCI60 database

### ***Cell-cell correlations on the basis of gene expression profiles (T-matrix)***

NCI has applied selective filters to reduce the initial 9,703 gene spots to a 1,376 genes that showed strong patterns of variation among the cell lines: cluster analyses, on the basis of gene expression pattern using a variety of algorithms and metrics organizes the cell lines into groups that reflect their tissue of origin (Fig. A - 4 (a)). With average linkage clustering and a correlation metric, the 1,376 genes, yielded 11 distinct cell clusters differing in average inter-cluster correlation coefficient by more than 0.3.

### ***Cell-cell correlations on the basis of drug activity profiles (A-matrix)***

From the overall database of more than 70,000 chemical compounds tested, NCI has selected for this analysis 1,400 compounds that has been tested at least four times on all or most of the 60 cell lines. Most of the drugs currently in clinical use are considered for cancer treatment.

The 60 cell lines have been originally clustered using an average-linkage algorithm and a metric based on the growth inhibitory activities (*GI50*) of the 1,400 compounds (Fig. A - 4 (b)): comparison of Fig. A - 4 (a) and Fig. A - 4 (b) indicates that the clustering by organ of origin was not as strong on the basis of activity as it was on the basis of gene expression. They observed 15 distinct branches at an average inter-cluster correlation coefficient of more than or equal to 0.3.

This difference in clustering (Fig. A - 4 (a, b)) was probably due, at least in part, to the activity of genes important to drug sensitivity and resistance. For example, several tumour cell lines known to express the multi-drug resistance gene ABCB1 (formerly MDR1) had closely related drug-activity profiles.

For quantitative comparison of the clustering (Fig. A - 4 (a, b)), they used the mean Pearson correlation coefficient  $p$  of all the Pearson correlation coefficients relating all possible pairs of cells in terms of their response to drugs and in terms of their gene expression. For these data sets,  $p$  was only 0.21. If these clustering had been identical,  $p$  would have been unity; if there had been no relationship at all,  $p$  would have been 0.

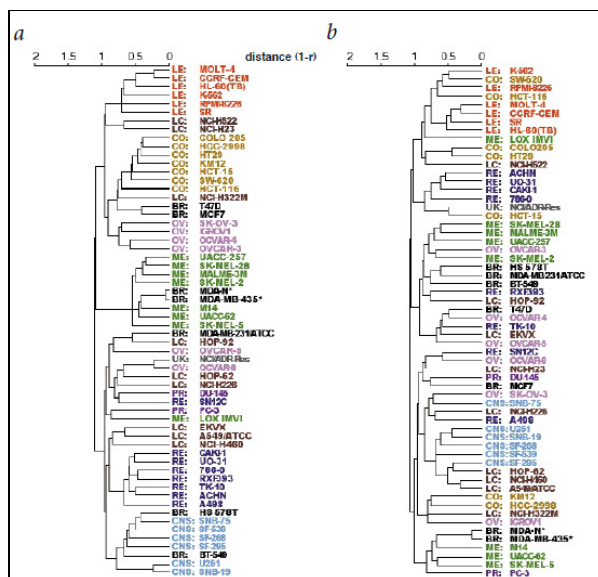


Fig. A - 4: dendrogram showing average-linkage hierarchical clustering of human cancer cell lines

The pharmacological implications of gene expression profiling studies of the NCI60 cell lines are of great importance. Because the gene expression patterns were determined in untreated cells, the data relates to sensitivity to therapy, rather than to the molecular consequences of therapy. In that sense, the study proposed in (Scherf et al., 2000) is analogous to an assessment of clinical tumours for markers that predict sensitivity to therapy. Their aim is to understand molecular pharmacology, to provide a rationale for selection of therapy on the basis of molecular characteristics of a patient's tumour. Subsection 5.2 would be shown how the relational approach can be more effective than traditional cluster analysis.

### A.3 Oral Anticoagulation Therapy (OAT) Data

Oral anticoagulation therapy data used came from two clinical studies we have been participated. In particular clinical and therapeutical data are collected from computerized databases of two clinical institutes in Milan: Istituto Auxologico and Clinica Humanitas.

In this way we build a database of 4000 patients of which around 380 have been so far genotyped. In the applications proposed in section 0 we considered only a subset of 1013 patients.

The collected data are imported in a database characterized by three entities: patients, therapy and visits.

In particular, for each patient we have information about date of birth, sex, medical evidence leading to OAT (Atrial Fibrillation, Deep Venous Thrombosis, other), patient's INR range (2-3, 2.5-3.5, 2.5-3).

Furthermore, for each patient, we memorize the concurrent medications in the therapy entity. In particular we classified all drugs in different categories: digitalis, amiodarone, furosemide, nitrates, beta blockers, calcium channel blockers, ACE Inhibitors, diuretic tiazidic, sartanic, farmaco lipids and other. So for each patient and for a particular category, we have a value "yes" if patient assumes a drug belong to this category and value "no" otherwise.

Finally, for each visit we collected the date of visit, the result of the INR measurement and the weekly dose and drug used for OAT therapy (Coumadin 5 mg, Sintrom 1 or 4 mg).

Relational structure of database is represented in Fig. A – 5. In particular, a *one to n* relation exists between patient and visit entities and an *n to n* relation exists between visit and therapy entities.

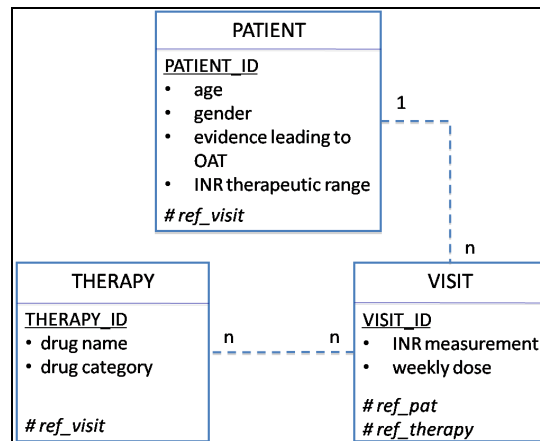


Fig. A – 5: Relational model of OAT application

For only a subset of patients (380) we collected in the patient entity genomic data. In particular the polymorphism of CYP2C9 and VKRC01 are collected. For each patient CYP2C9 gene feature can have the following values: WT (wild Type), AC, CC. The possible values for gene VKORC1 are: WT (wild type), CT and TT.

For a better description of each polymorphism we remand to (Anderson et al., 2007).

Entry characteristics for both 1013 patients and the subset of 135 patients with genomic data are summarized in Tab. A - 1:

Characteristics	Patient without genomic data	Patient with genomic data
Patients number	1013	380
Age, y, mean (dev.std)	76 (10)	76 (11)
<b>Gender:</b>		
Women N (%)	502 (49.5%)	180 (47.40%)
Men N (%)	511 (50.44%)	200 (52.60%)
<b>Primary reason for anticoagulation, N (%)</b>		
Atrial fibrillation	771(76.11 %)	360 (94.70%)
Deep vein thrombosis	80 (7.9%)	15(3.9 %)
Other diagnosis	162 (15.99 %)	5 (1.4%)
<b>Clinical Variables:</b>		
Takes amiodarone, N (%)	175 (17.20%)	66 (17,36%)
Takes ASA (acetylsalicylic acid), N (%)	110 (10.85%)	35 (9.2%)
Takes Farmaco Lipids, N (%)	213 (21.02%)	49 (12.9%)

**Tab. A - 1:** OAT patients' characteristics

The sample shows a prevalence of atrial fibrillation (76.11%). The genotyped sub-sample mirrors in a balanced way the relative weight of the features in the large one. In our studies we extract from the 380 patients, only those with atrial fibrillation and so we work on a dataset of 360 patients.

The allelic variant frequency for the subset of 360 patients is summarized in the table below.

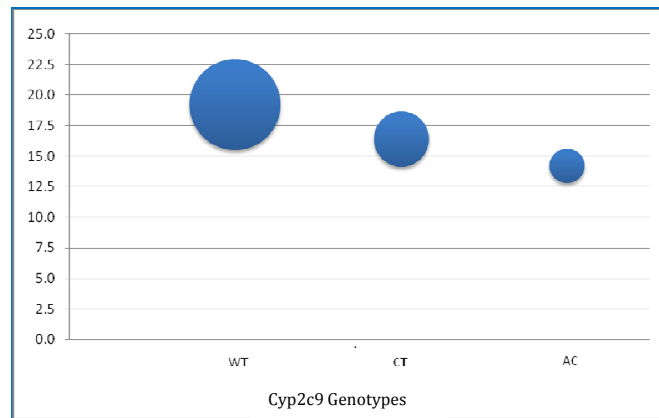
Allelic variant frequencies		
<b>CYP2C9</b>	<i>WT</i>	66.67%
	<i>CT</i>	20.74%
	<i>AC</i>	12.59%
<b>VKORC1</b>		
	<i>WT</i>	40.74%
	<i>CT</i>	33.33%
	<i>TT</i>	25.93%

**Tab. A - 2:** Allelic variant frequencies of gene CYP2C9 and VKORC1

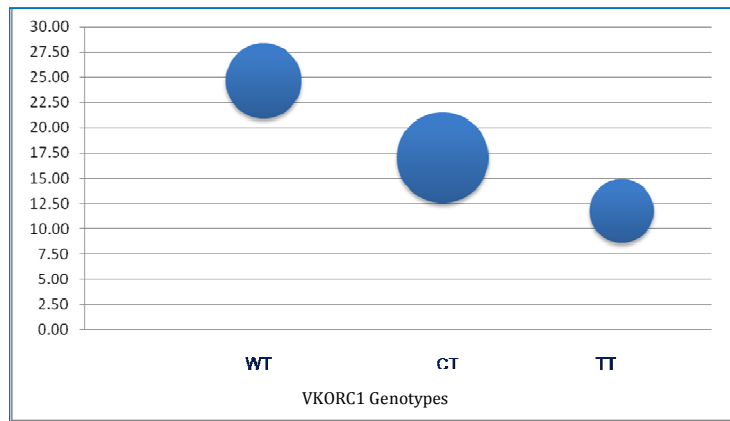
The overall allelic frequency distribution is similar to what is reported in the literature (Anderson et al., 2007; Gage et al., 2008).

In the following figures (Fig. A - 6 and Fig. A - 7) we report the genotypes prevalence for both genes and the mean weekly maintenance dosing for Warfarin. Circle represents relative population size.





**Fig. A - 6:** CYP2C9 genotypes prevalence and the mean weekly maintenance dosing for Warfarin.



**Fig. A - 7:** VKORC1 genotypes prevalence and the mean weekly maintenance dosing for Warfarin.

As we can see the weekly maintenance dose for patients with polymorphism is less than that for wild type patients.

## **A.4 Oncological Data**

Genetic Programming as feature selection method has been tested on two publicly available oncologic datasets: the first one contains data from healthy colon tissues and colon tissues affected by cancer and will be called *Colon Dataset* from now on; the second one contains data from patients affected by two different kinds of leukemia (acute myeloid leukemia and acute lymphoblastic leukemia) and will be called *Leukemia Dataset* from now on. These two datasets are described as follows.

### **A.4.1 Colon Dataset**

The Colon Dataset is a collection of expression measurements from colon biopsy samples reported in (Alon et al., 1999).

The dataset consists of 62 samples of colon epithelial cells collected from colon-cancer patients. In particular the “tumour” biopsies were extracted from tumours, and the “normal” biopsies were collected from healthy parts of the colons of the same patients. The final assignments of the status of biopsy samples were made by pathological examination.

Gene expression levels in these 62 samples were measured using high-density oligonucleotide arrays. Of the about 6000 genes represented in these arrays, 2000 genes were selected based on the confidence in the measured expression levels. The dataset, 62 samples over 2000 genes, is available at <http://microarray.princeton.edu/oncology/affydata/index.html>.

### **A.4.2 Leukemia Dataset**

The Leukemia Dataset (first introduced in (Golub et al., 1999)) contains data from 72 patients, half of which affected by acute myeloid leukemia and the remaining ones affected by lymphoblastic leukemia.

For these patients, 7070 genes have been monitored. For measuring the expression level of those genes, oligonucleotides microarrays produced by Affimetrix have been used. Thus, the dataset is composed by 7070 columns and 72 lines, each of which labelled with “myeloid” or “lymphoblastic” in order to separate these two kinds of leukemia.

This dataset and a detailed description of it can be found at:

<http://genecruiser.broadinstitute.org/cgi-bin/cancer/publications/pubpaper.cgi?mode=view&paper id=43>.

### **A.4.3 Molecular Dataset**

The molecular dataset is built by a collaboration with Delos s.r.l.

In particular, a small set of estrogen-genistein virtual molecules have been collected from the RCSB PDB database (RCSB Protein Data Bank, 2007). Successively substitution points on which we have clasped a small database of substituents (OH, CH<sub>3</sub>, CH<sub>2</sub>CH<sub>3</sub>, CH<sub>2</sub>OH, CH<sub>2</sub>CH<sub>2</sub>OH, CH<sub>2</sub>CH<sub>2</sub>NH<sub>2</sub>, OCH<sub>2</sub>CH<sub>2</sub>NH<sub>2</sub>) have been defined, obtaining a set of 992 genistein based virtual molecules.

The resulting chemical structures were then optimized by means of molecular mechanics using the MOE software (Molecular Operating Environment (MOE), 2007) and MMFF94 force field (MMFF94, 2007) for calculating 267 molecular descriptors.

Finally, for each one of these ligands, we have calculated their docking energy value by means of the DELOS software platform (Delos s.r.l., 2007), an environment for effective virtual screening and docking simulations recently produced by the Discovery and Lead Optimization Systems company (Bresso, Italy).

The resulting dataset was composed of 992 genistein based molecules, each of which is represented by a vector of 267 molecular descriptors and with known values of the docking energy. It can be downloaded from the web page: <http://personal.disco.unimib.it/Vanneschi/Docking.htm>.

The dataset is a matrix  $H=[H_{(i,j)}]$  of 992 rows and 268 columns, where each line  $i$  represents a molecule whose known docking energy value has been placed at position  $H_{(i,268)}$ .

In this way, the last column of matrix  $H$  represents all the known docking energy values.

## Bibliography

Abasolo D., Hornero R., Espino P., Alonso A., de la Rosa R., Electroencephalogram analysis with approximate entropy to help in the diagnosis of Alzheimer's disease, *in: Proceedings of the 4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, Birmingham, U.K.*, pp. 222–225, 2004.

Aebersold R., Hood L.E., and Watts J.D. Equipping scientists for the new biology. *Nat. Biotechnol.* Vol. 18, pp. 359, 2000.

Ahmad A. and Dey L., A k-mean clustering algorithm for mixed numeric and categorical data, *Data & Knowledge Engineering*, Vol. 63, Num. 2, pp. 503-527, 2007.

Akaike H., Information theory and an extension of maximum likelihood principle, *In 2nd International Symposium on Information Theory*, Akademia Kiado, June, 1973.

Alon U., Barkai N., Notterman D. A., Gish K., Ybarra S., Mack D., Levine A.J., Broad patterns of gene expression revealed by clustering analysis of tumour and normal colon tissues probed by oligonucleotide arrays, *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 12, pp. 6745–6750, 1999.

Alqadah F., Bhatnagar R., 'An effective algorithm for mining 3-clusters in vertically partitioned data', in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge mining*, pp. 1103-1112, 2008.

Alter O., Brown P.O. and Botstein D. Singular value decomposition for genome-wide expression data processing and modelling, *PNAS*, Vol. 97, pp. 10101–10106, 2000.

Anderberg, M. R., Cluster Analysis for Applications. *Academic Press*, New York, December 1973.

Anderson J.L., Horne B.D., Stevens S.M., Grove A.S., Barton S., Nicholas Z.P., Kahn S.F., May H.T., Samuelson K.M., Muhlestein J.B., Carlquist J.F., Randomized trial of genotype-guided versus standard warfarin dosing in patients initiating oral anticoagulation, *Circulation*; Vol. 116, No.22, pp.2563-70, 2007.

Andreopoulos B., An A, Wang X. Bi-Level Clustering of mixed categorical and numerical biomedical data, *Int Journal of DataMining in Bioinformatics*, Vol. 1, pp. 19–56, 2006.

Andreopoulos B., An A, Wang X. and Schroeder M., "A roadmap of clustering algorithms: finding a match for biomedical application", *Briefing in Bioinformatics*, Vol. 10, No. 3 pp. 297-314, 2009.

Archetti F., Campanelli P., Fersini F., Messina E., A Hierarchical Document Clustering Environment Based on the Induced Bisecting k-Means *In Proceeding of the 7th International Conference on Flexible Query Answering Systems*, 2006.

Archetti F., Fersini E., Giordani I., Mauri G., Messina E., Relational clustering in the inference of genetic regulatory networks, *Mini EURO Conference on Computational Biology, Bioinformatics and Medicine*, September 2008.

Archetti F., Giordani I., Vanneschi L., Genetic Programming for Anticancer Therapeutic Response Prediction using the NCI-60 Dataset, *Computers and Operations Research*, in press. 2009a

Archetti F., Giordani I., Vanneschi L., Genetic programming for QSAR investigation of docking energy, *Applied Soft Computing*, in press, 2009b

- Archetti F., Giordani I., Mauri G., Messina E., A new clustering approach for learning transcriptional regulatory modules, *Proceedings of BITS09*, Sixth Annual Meeting Bioinformatic Italian Society, March 18-20 Genoa, pp:76-77, 2009c.
- Archetti F., Giordani I., Messina E., Ogliari G., Mari D., "A comparison of data mining approaches in the categorization of oral anticoagulant patients", to appear in the *International Workshop of Applications of Machine Learning in Bioinformatics (satellite workshop of IEEE International Conference on Bioinformatics and Biomedicine-BIBM-)*, November, 2009d
- Archetti F., Giordani I., Messina E., Ogliari G., Mari D., "A Markov based classification and treatment of the anticoagulant patient", *In proceedings of AIRO 2009, Annual Conference of the Italian Operations Research Society*, September, 2009e
- Baeza-Yates R. and Ribeiro-Neto B., *Modern Information Retrieval*. ACM Press, New York, 1999.
- Bailey J.E., Lessons from metabolic engineering for functional genomics and drug discovery, *Nat. Biotechnol.*, Vol. 17, pp. 616–618, 1999.
- Ball G. and Hall D., A Clustering Technique for Summarizing Multivariate Data. *Behavior Science*, Vol. 12, pp. 153–155, March 1967.
- Banerjee A., Dhillon I., Ghosh J., Sra S., Generative model-based clustering of directional data. *In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pp. 19–28, 2003.
- Banerjee, A. , Merugu S. I., Dhillon, S. and Ghosh J. Clustering with Bregman Divergences. *In Proc. of the 2004 SIAM Intl. Conf. on Data Mining*, pp. 234–245, 2004.
- Bansal N., Blum A., Chawla S., Correlation clustering. *In Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS-02)*, pp. 238–247, 2002.
- Banzhaf W., Nordin P., Keller R. E., and Francone F. D., *Genetic Programming: An Introduction*. New York: Morgan Kaufmann, 1998.
- Barash Y., Bejerano G., and Friedman N., "A Simple Hyper-Geometric Approach for Discovering Putative Transcription Factor Binding Sites" *In Proceedings of the First international Workshop on Algorithms in Bioinformatics* (August 28 - 31, 2001), Lecture Notes In Computer Science, Vol. 2149, pp. 278-293, 2001.
- Bar-Hillel A., T. Hertz, N. Shental, and D. Weinshall. Learning a Mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, Vol. 6, pp. 937-965, 2005.
- Basu S., Banerjee A., and Mooney R. J., Semi-supervised clustering by seeding. *In Proceedings of 19th International Conference on Machine Learning (ICML-2002)*, pp. 19–26, 2002.
- Basu S., Banerjee A., and Mooney R. J., Active semi-supervision for pairwise constrained clustering. *In Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04)*, 2004.
- Basu S., Bilenko S., and Mooney R. J., A probabilistic framework for semi-supervised clustering. *In KDD04*, pp. 59–68, 2004.
- Ben-Bassat M., Pattern recognition and reduction of dimensionality. *Handbook of Statistics II*—Krishnaiah P, Kanal L, eds. Vol. 1, pp. 773–791, 1982.
- Ben-Dor A., Friedman N. and Yakhini Z. Class discovery in gene expression data. *RECOMB*, pp. 31–38, 2001.
- Ben-Dor A., Chor B., Karp R., and Yakhini Z. Discovering local structure in gene expression data: The order-preserving submatrix problem. *In Proceedings of the 6th*

- International Conference on Computational Biology (RECOMB'02)*, pp. 49–57, 2002.
- Bergmann S., Ihmels J. and Barkai N., “Iterative signature algorithm for the analysis of largescale gene expression data”, *Phys Rev E Stat Nonlin Soft Matter Phys*, Vol. 67, 2003.
- Bilenko M. and Mooney R. J., Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pp. 39–48, 2003.
- Bishop C., *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.
- Blum A. and Mitchell T., Combining labelled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100, 1998.
- Boley, D. Principal Direction Divisive Partitioning. *Data Mining and Knowledge Discovery*, Vol. 2, No.4, pp.325–344, 1998.
- Bradley, P. S. and Fayyad, U. M. Refining Initial Points for K-Means Clustering. In *Proc. of the 15th Intl. Conf. on Machine Learning*, pages 91–99, 1998.
- Burioka N., Corn´ elissen G., Halberg F., Kaplan D.T., Suyama H., Sako T., Shimizu E., Approximate entropy of human respiratory movement during eye-closed waking and different sleep stages, *Chest*, Vol. 123, pp. 80–86, 2003.
- Bushel P. R., Wolfinger R.D., Gibson G., Simultaneous clustering of gene expression data with clinical chemistry and pathological evaluations reveals phenotypic prototypes, *BMC Systems Biology*, pp. 1-15, 2007.
- Carney M., Cunningham P., The Benefits of Using a Complete Probability Distribution when Decision Making: An Example in Anticoagulant Drug Therapy, Trinity College Dublin, Department of Computer Science, *Technical Report*, pp.22. 2005
- Casillas J., Cordon O., Del Jesus M. J., and Herrera F., Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems, *Inform. Sci.*, Vol. 136, pp. 135-157, 2001.
- Chang J.H., Hwang K.B., Zhang B.T., “Analysis of gene expression profiles and drug activity patterns by clustering and Bayesian network learning”, In *Methods of Microarray Data Analysis II*, chapter 11. Edited by Lin SM, Johnson KF. Kluwer Academic Publishers, pp. 169-184, 2002.
- Chang, H., & Yeung, D.-Y., Locally linear metric adaptation for semi-supervised clustering. In *Proceedings of 21st International Conference on Machine Learning (ICML) 2004*.
- Chang J.H., Hwang K.B., Oh S.J., Zhang B.T., “Bayesian network learning with feature abstraction for gene-drug dependency Analysis”. *J Bioinform Comput Biol*, Vol. 3, No.1, pp. 61-77, 2005.
- Cheng Y. and Church G.M., Biclustering of expression data. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB'00)*, pp. 93–103, 2000.
- Cherry J.M., Adler C., Ball C., Chervitz S.A., Dwight S.S., Hester E.T., Jia Y., Juvik G., Roe T., Schroeder M., Weng S., Botstein D., “Sgd: Saccharomyces genome database”, *Nucleic Acid Res.*, Vol. 26, pp. 73–79, 1998.
- Chi Y., Muntz R. R., Nijssen S., and Kok J. N.. Frequent subtree mining - an overview. *Fundamenta Informaticae*, Vol. 66, No.1-2, pp. 161–198, 2005.
- Clements M., Van Somerena E. P., Knijnenburga T. A. and Reindersa M. J.T., “Integration of Known Transcription Factor Binding Site Information and Gene Expression Data to

- Advance from Co-Expression to Co-Regulation" *Genomics, Proteomics & Bioinformatics*, Vol. 5, No. 2, pp. 86-101, 2007.
- CLUTO 2.1.1: Software for Clustering High-Dimensional Datasets. [www.cs.umn.edu/karypis](http://www.cs.umn.edu/karypis), 2003.
- Cohn D., Caruana R., McCallum A., Semi-supervised clustering with user feedback. *Technical Report TR2003-1892*, Cornell University, 2003.
- Datta S., Giannella C., and Kargupta H., K-Means Clustering over a Large, Dynamic Network. In *Proceedings of 2006 SIAM Conference on Data Mining*, Bethesda, MD, April 2006.
- Datta S. and Kargupta H., Uniform Data Sampling from a Peer-to-Peer Network. In *2007 IEEE International Conference on Distributed Computing Systems (ICDCS 2007)*, Toronto, Canada, June, 2007.
- DELOS S.r.l Discovery and Lead Optimization Systems, 20091, Bresso (MI), Italy, 2007. <http://www.delos-bio.it>.
- Demiriz A. , Bennett K. P., and Embrechts M. J., Semi-supervised clustering using genetic algorithms. In *Artificial Neural Networks in Engineering (ANNIE-99)*, pp. 809-814, 1999.
- Dhillon I. S. and Modha D. S., Concept decompositions for large sparse text data using clustering. *Machine Learning*, Vol. 42, pp. 143-175, 2001.
- Dhillon, I. S., Guan Y. and Kogan J. Iterative Clustering of High Dimensional Text Data Augmented by Local Search. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 131-138, 2002.
- Ding C.H.Q., Unsupervised Feature Selection Via Two-way Ordering in Gene Expression Analysis, *Bioinformatics*, Vol. 19, pp. 1259-1266, 2003.
- Duda R.O., Hart P.E., and Stork D.G., *Pattern Classification*, John Wiley & Sons, New York, 2000.
- Dutta H., Giannella C., Borne K. and Kargupta H., Distributed Top-K Outlier Detection from Astronomy Catalogs using the DEMAC System. *Proceedings of the SIAM International Conference on Data Mining*, Minneapolis, USA, 2007.
- Eckman M., Rosand J., Greenberg S., and Gage B., "Cost-effectiveness of using pharmacogenetic information in warfarin dosing for patients with nonvalvular atrial fibrillation," *Annals of Internal Medicine*, vol. 150, no. 2, pp. 73-83, 2009.
- Eisen M.B., Spellman P.T., Brown P.O. and Botstein D., "Cluster analysis and display of genome-wide expression patterns", *Proc. Natl. Acad. Sci.*, pp. 14863-14868, 1998.
- Ernst J., Beg Q. K., Kay K. A., Balázsi G., Oltvai Z. N. and Bar-Joseph Z., "A semi-supervised method for predicting transcription factor-gene interactions in escherichia coli" *PLoS computational biology* , Vol. 4, No. 3, 2008.
- Evans G.A. Designer science and the "omic" revolution. *Nat. Biotechnol.* Vol. 18, pp. 127 2000.
- Ferri F., Pudil P., Hatef M. and Kittler J., *Pattern Recognition in Practice IV, Multiple Paradigms, Comparative Studies and Hybrid Systems*, Elsevier, pp.403-413, 1994.
- Fersini E., Giordani I., Messina E., Archetti F., Discovering Relationships Among Human Cancer, Gene Expression Profile and Drug Responses: A Relational Clustering Approach, *to appear in the Sysbiohealth Symposium 2009*, November 2009a.
- Fersini E., Giordani I., Messina E., Archetti F., "Relational Clustering and Bayesian Networks for Linking Gene Expression Profiles and Drug Activity Patterns", *to appear in*

*the International Workshop of Applications of Machine Learning in Bioinformatics (satellite workshop of IEEE International Conference on Bioinformatics and Biomedicine-BIBM-),* November 2009b.

Fisher D. Iterative Optimization and Simplification of Hierarchical Clusterings. *Journal of Artificial Intelligence Research*, Vol. 4, pp.147–179, 1996.

Gage B.F., Use of pharmacogenetic and clinical factors to predict the therapeutic dose of fofari, *Clin Pharmacol Ther*, Vol. 84, No. 3, pp. 326-31, 2008.

Ganti V., Gekhre J.E., Ramakrishnan R., CACTUS-clustering categorical data using summaries, in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 73–83, 1999

Garcia D., Ageno W., Bussey H., Eikelboom J., Margaglione M., Marongiu F., Moia M., Palareti G., Pengo V., Poli D., Schulman S., Witt D., Wittkowsky A., and Crowther M., “Prevention and treatment of bleeding complications in patients receiving vitamin k antagonists, part 1: Prevention.” *American journal of hematology*, vol. 84, no. 9, pp. 579–583, 2009.

Gasch A.P., Spellman P.T., Kao C.M., Carmel-Harel O., Eisen M.B., Storz G., Botstein D. and Brown P.O. “Genomic expression program in the response of yeast cells to environmental changes” *Mol. Biol. Cell*, Vol. 11, pp. 4241–4257, 2000.

GenBank. The National Institutes of Health (NIH) genetic sequence database, an annotated collection of all publicly available DNA sequences, 2008. See <http://www.ncbi.nlm.nih.gov/Genbank/>.

Getz G., Levine E., and Domany E. Coupled two-way clustering analysis of gene microarray data. In *Proceedings of the Natural Academy of Sciences USA*, pp. 12079–12084, 2000.

Gluck M., Corter J., Information, uncertainty, and the utility of categories. *Proc 7th Ann Conf Cog Soc*, pp.283-287, 1985

Goldberg, D. E., Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

Golub T. R., Slonim D. K., Tamayo P., “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring,” *Science*, pp. 531–537, 1999.

Gonzalez T.F., Clustering to minimize the maximum inter-cluster distance, *Journal of Theoretical Computer Science*, Vol. 38, pp. 293-306, 1985.

Goodall D.W., A new similarity index based on probability, *Biometric*, Vol. 22, pp. 882–907, 1996.

Graepel T., Burger M., Obermayer K., “Self-organizing maps: generalizations and new optimization techniques” *Journal of Neurocomputing*, Vol. 21, pp.173-190, 1998.

Guha S., Rastogi R. and Shim K., ‘ROCK: A Robust Clustering Algorithm for Categorical Attributes’ in *Proc. of ICDE ’99* pp. 512-521, 1999.

Guyon I., Elisseeff A., An introduction to variable and feature selection, *J. Mach Learn Res*. Vol. 3, pp. 1157–1182, 2003.

Guyon I., Weston J., Barnhill S. and Vapnik V., Gene selection for cancer classification using support vector machines. *Mach. Learn.* Vol. 46, pp. 389–422, 2002

Hall M., Correlation-based feature selection for machine learning, *PhD Thesis New Zealand: Department of Computer Science, Waikato University*, 1999.



- Han J. and Kamber M., *Data mining: Concepts and Techniques*, 2<sup>nd</sup> edition, Morgan Kaufmann, 2006.
- Hand D.J., Mannila H., Smyth P., *Principles of Data Mining*. *The MIT Press*, 2001.
- Hartigan J. A. Direct clustering of a data matrix. *Journal of the American Statistical Association (JASA)*, Vol. 67, No. 337, pp. 123–129, 1972.
- Hartwell L.H., Leibler S. and Murray A.W. From molecular to modular cell biology. *Nature*, Vol. 402, pp. C47–C52, 1999.
- Hastie T., Tibshirani R., Eisen M., Alizadeh A., Levy R., Staudt L., Chan W., Botstein D. and Brown P., 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns, *GenomeBiology*, 2000.
- Haverty P.M., Hansen U. and Weng Z. "Computational inference of transcriptional regulatory networks from expression profiling and transcription factor binding site identification", *Nucleic Acids Res*, Vol. 32, pp. 179-188, 2004.
- He Z., Xu X., Deng S., Scalable algorithms for clustering large datasets with mixed type attributes, *International Journal of Intelligence Systems*, Vol. 20, pp. 1077–1089, 2005.
- He Z., Xu X., Deng S., Squeezer: An efficient algorithms for clustering categorical data, *Journal of Computer Science and Technology*, Vol. 17, No. 5, pp. 611–624, 2002
- Hereford L.M., Osley M.A., Ludwig T.R.D. and McLaughlin C.S., Cell-cycle regulation of yeast histone mRNA. *Cell*, Vol. 24, pp. 367–375, 1981.
- Herrero J., Diaz-Uriarte R. and Dopazo J. Gene expression data preprocessing, *Bioinformatics*, Vol. 19, pp. 655–656, 2003.
- Hertz T., Bar-Hillel A., Weinshall D., Boosting margin based distance functions for clustering. *In Proceedings of 21st International Conference on Machine Learning (ICML)*, 2004.
- Ho K., Moody G., Peng C., Mietus J., Larson M., Levy D., Goldberger A., Predicting survival in heart failure case and control subjects by use of fully automated methods for deriving nonlinear and conventional indices of heart rate dynamics, *Circulation* Vol. 96, No. 3, pp. 842–848, 1997.
- Holland J., *Adaptation in Natural and Artificial Systems*, *Ann Arbor: University of Michigan Press*, 1975.
- Holter N.S., Mitra M., Martian A., Cieplak M., Banavar J.R. and Fedoroff N.V. Fundamental patterns underlying gene expression profiles. *PNAS*, Vol. 97, pp. 8409–8414, 2000.
- Huang Z., Clustering large data sets with mixed numeric and categorical values, *in: Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, World Scientific, Singapore, 1997.
- Huang Z., Extensions to the K-modes algorithm for clustering large data sets with categorical values, *Data Mining and Knowledge Discovery*, Vol. 3, 1998.
- Huang Z., Ng M.K., Rong H., Li Z., Automated variable weighting in k-mean type clustering, *IEEE Transactions on PAMI*, Vol. 27, No. 5, 2005.
- Ihmels J., Bergmann S. and Barkai N., "Defining Transcription Modules Using Large Scale Gene Expression Data" *Bioinformatics*, Vol. 20, pp. 1993–2003, 2004.
- Inza I, Etxeberria R., Sierra B., Feature subset selection by Bayesian networks based optimization. *Artif. Intell.* Vol. 123, pp. 157–184, 2000.
- Jain A. K. and Dubes R. C., *Algorithms for Clustering Data*. Prentice Hall Advanced Reference Series. *Prentice Hall*, March 1988.

- Jain A., Duin R., and Mao J., "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 22, No. 1, pp. 4–37, 2000.
- Januzaj E., Kriegel H.-P., and Pfeifle M., Scalable Density-Based Distributed Clustering. *In The 15<sup>th</sup> European Conference on Machine Learning (ECML) and the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Pisa, Italy, September 2004.
- Jardine N. and Sibson R. *Mathematical Taxonomy*. Wiley, New York, 1971.
- Joachims T., Transductive inference for text classification using support vector machines. *In Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)*, pp. 200–209, 1999.
- Karypis G., Han E.-H., and Kumar. Multilevel Refinement for Hierarchical Clustering. *Technical Report TR 99-020, University of Minnesota, Minneapolis, MN*, 1999
- Kaufman L. and Rousseeuw P.J., *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley & Sons, 1990.
- Keijzer M. Improving symbolic regression with interval arithmetic and linear scaling. *Genetic Programming, Proceedings of EuroGP'2003*, Vol. 2610, pp. 70-82, 2003.
- Keijzer M. Scaled symbolic regression. *Genetic Programming and Evolvable Machines*, Vol. 5, No. 3, pp. 259–269, 2004.
- Kittler J., *Pattern Recognition and Signal Processing*, Chapter Feature Set Search Algorithms Netherlands: Sijthoff and Noordhoff, Alphen aan den Rijn. pp. 41–60, 1978.
- Klein D., Kamvar S. D., and Manning C., From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. *In Proceedings of the The Nineteenth International Conference on Machine Learning (ICML-2002)*, pp. 307–313, 2002.
- Klugar Y., Basri R., Chang J.T., and Gerstein M. Spectral biclustering of microarray data: coclustering genes and conditions. *In Genome Research*, volume 13, pages 703–716, 2003.
- Knorr E., Ng R., Tucakov T., "Distance-based Outliers: Algorithms and Applications," *VLDB Journal*, Vol. 8, No. 3-4, pp. 237-253, 2000.
- Koller D, Sahami M., Toward optimal feature selection, *Proceedings of the Thirteenth International Conference on Machine Learning* Bari, Italy. pp. 284–292, 1996.
- Koza J. R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992
- Kudo M. and Sklansky J., Comparison of algorithms that select features for pattern classifiers, *Patt. Recognit.*, Vol. 33, pp. 25-41, 2000.
- Kutalik Z., Beckmann J. and Sven B., "A modular approach for integrative analysis of large-scale gene-expression and drug response data", *Nat Biotech*, Vol. 26, pp. 531-539, 2008.
- Langdon W. B. and Banzhaf W. Repeated patterns in tree genetic programming. *In M. Keijzer et al., editor, Genetic Programming, 8th European Conference, EuroGP2005*, Vol. 3447 of Lecture Notes in Computer Science, pp. 190–202, 2005.
- Langdon W. B. and Poli R. *Foundations of Genetic Programming*. Springer, Berlin, Heidelberg, New York, Berlin, 2002.
- Lazzeroni L. and Owen A. Plaid models for gene expression data. *Technical report, Stanford University*, 2000.

- Lee T.I., Rinaldi N.J., Robert F., Odom D.T., Bar-Joseph Z., Gerber G.K., Hannett N.M., Harbison C.T., Thompson C.M. and Simon I. "Transcriptional regulatory networks in *Saccharomyces cerevisiae*" *Science*, Vol. 298, pp. 799-804, 2002.
- Lesko L, The critical path of warfarin dosing: finding an optimal dosing strategy using pharmacogenetics, *Clin Pharmacol Ther*, Vol. 84, No.3, pp. 301-303, 2008
- Levy W., Pantin E., Mehta S., McGarvey M., Hypothermia and the approximate entropy of the electroencephalogram, *Anesthesiology*, Vol. 98, No.1, pp. 53-57, 2003.
- Li C., Biswas G., Unsupervised learning with mixed numeric and nominal data, *IEEE Transactions on Knowledge and Data, Engineering*, Vol. 14, No.4, pp. 673-690, 2002.
- Li S, Lee R., Lang S.-D., "Mining Distance-Based Outliers from Categorical Data", *In proc. of Seventh IEEE International Conference on Data Mining Workshops*, pp.225-230, 2007.
- Liu H., Li J. and Wong L., A Comparative Study on Feature Selection and Classification Methods Using Gene Expression Profiles and Proteomic Patterns. *In Proceedings of 13th International Conference on Genome Informatics*, pp. 51-60, 2002.
- Liu H., Motoda H., Feature Selection for Knowledge Discovery and Data Mining, *Norwell, MA: Kluwer Academic Publishers*, 1998.
- Liu X., Brutlag D.L. and Liu J.S., "Bioprospector: discovering conserved dna motifs in upstream regulatory regions of co-expressed genes" *In proceedings Pacific Symposium on Biocomputing*, pp. 127-38, 2001.
- Luo H., Kong F., Li Y., Clustering mixed data based on evidence accumulation, *in: X. Li, O.R. Zaiane, Z. Li (Eds.), ADMA 2006, Lecture Notes on Artificial Intelligence 4093*, 2006.
- Mack G.S., Can complexity be commercialized? *Nature Biotechnology*, Vol. 22, pp. 1223 - 1229, 2004.
- MacQueen J., Some methods for classification and analysis of multivariate observations. *In Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, pp. 281-297, 1967.
- Madeira S. C. and Oliveira A. L., Biclustering algorithms for biological data analysis: a survey, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol.1, pp. 24-45, 2004.
- Mager W.H., and De Kruijff A.J., Stress-induced transcriptional activation. *Microbiol. Rev.* 59, 506-531, 1995.
- Mannis G., Fast computation of approximate entropy, *Computer methods and programs in biomedicine*, Vol. 91, pp. 48-54, 2008.
- Matys V., Kel-Margoulis O. V., Fricke E., Liebich I., Land S., Barre-Dirrie A., Reuter I., Chekmenev D., Krull M., Hornischer K., Voss N., Stegmaier P., Lewicki-Potapov B., Saxel H., Kel A. E. and Wingender E., "Transfac(r) and its module transcompel(r): transcriptional gene regulation in eukaryotes" *Nucleic Acids Res*, Vol. 34, pp.108-110, 2006.
- McAdams H.H. and Arkin A. Simulation of prokaryotic genetic circuits. *Annu. Rev. Biophys. Biomol. Struct.* Vol. 27, pp. 199-224, 1998.
- McDonald S., Xydeas C., Angelov P., A Retrospective Comparative Study of three Data Modelling Techniques in Anticoagulation Therapy, *International Conference on BioMedical Engineering and Informatics BMEI2008*, Sanya, Hainan, China, pp. 219-225, 2008
- Metz C. E., "Basic principles of ROC analysis," *Seminars in Nuclear Medicine*, Vol. 8, No. 4, pp. 283-298, 1978.

- Middendorf M., Kundaje A., Shah M., Freund Y., Wiggins C. H. and Leslie, C., "Motif discovery through predictive modeling of gene regulation" *In Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB 2005)*, Cambridge, USA, 2005.
- Mirkin B., Mathematical Classification and Clustering. *Kluwer Academic Publishers*, 1996.
- MMFF94 Validation Suite Created by Computational Chemistry list Ltd., 2007. <http://www.ccl.net/cca/data/MMFF94> .
- Molecular Operating Environment (MOE) A software developed by chemical computing group inc., 2007. <http://www.chemcomp.com> .
- Murali T. M. and Kasif S. Extracting conserved gene expression motifs from gene expression data. In *Proceedings of the Pacific Symposium on Biocomputing*, Vol.8, pp. 77-88, 2003.
- NCBI. The National Center for Biotechnology Information Database, 2008. See <http://www.ncbi.nlm.nih.gov/>.
- NCI60 Cancer Microarray Project. National Cancer Institute, Bethesda MD, 2008. See <http://genome-www.stanford.edu/nci60/>.
- Nigam K., McCallum A. K. , Thrun S. , and Mitchell T., Text classification from labeled and unlabeled documents using EM. *Machine Learning*, Vol. 39, pp. 103-134, 2000.
- Friedman N., Getoor L., Koller D., and Pfefier A., Learning probabilistic relational models. *In Proc. of the 16th International Joint Conference on Artificial Intelligence*, pp. 1300-1309. Morgan Kaufmann Publishers Inc., 1999.
- Pal N. R., Nandi S., and Kundu M. K., Self-crossover: A new genetic operator and its application to feature selection, *Int. J. Syst. Sci.*, Vol. 29, No. 2, pp. 207-212, 1998.
- Palsson B.O. What lies beyond bioinformatics? *Nat.Biotechnol.* Vol. 15, pp. 3-4, 1997.
- Palsson B.O., The challenges of in silico biology. *Nat Biotechnol.*, Vol. 18, pp. 1147-1150, 2000.
- Paull K.D., Shoemaker R.H., Hodes L., Monks A., Scudiero D.A., Rubinstein L., Plowman J., Boyd M.R., Display and analysis of patterns of differential activity of drugs against human tumor cell lines: development of mean graph and COMPARE algorithm. *J. Natl. Cancer Inst.*, Vol. 81, pp.1088-1092, 1989.
- Pelleg D. and Moore A. W. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. *In Proc. of the 17th Intl. Conf. on Machine Learning*, pp. 727-734. Morgan Kaufmann, San Francisco, CA, 2000.
- Pincus S., Approximate entropy as a measure of system complexity, *Proc. Natl. Acad. Sci. U.S.A.*, Vol. 88, No. 6, pp. 2297-2301, 1991.
- Pincus S., Approximate entropy (ApEn) as a complexity measure, *Chaos*, Vol. 5, pp. 110-117, 1995.
- Price C., Nasmyth K., and Schuster T., A general approach to the isolation of cell cycle-regulated genes in the budding yeast, *Saccharomyces cerevisiae*. *J. Mol. Biol.* 218, 543-556, 1991.
- Procopiuc C. M., Jones M., Agarwal P. K., Murali T. M. , A Monte Carlo algorithm for fast projective clustering. *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pp. 418-427, 2002.
- RCSB Protein Data Bank (PDB) An information portal to biological macromolecular structures, 2007. <http://www.rcsb.org/pdb/home/home.do> .

- Rosove M. H. and Grody W., "Should we be applying warfarin pharmacogenetics to clinical practice? no, not now," *Annals of Internal Medicine*, Vol. 151, No. 4, pp. 270–273, 2009.
- Roskopf M., Schmidt H. A., Feldkamp U., and W. Banzhaf, "Genetic programming based DNA microarray analysis for classification of tumour tissues," *Tech. Rep. 2007-03, Memorial University of Newfoundland*, 2007.
- Rousseeuw P. J., Leroy A.M., *Robust Regression and Outlier Detection*, Wiley, New York, 1987.
- Ruis, H., and Schuller, C., Stress signaling in yeast. *Bioessays* Vol. 17, pp. 959–965, 1995.
- Ryan et al., editor, Genetic Programming, *Proceedings of the 6th European Conference, EuroGP 2003*, Vol. 2610 of LNCS, pp. 71–83, 2003.
- Ryan S., Goldberger A., Pincus S., Mietus J., Lipsitz L., Gender and age-related differences in heart rate dynamics: are women more complex than men, *J. Am. Coll. Cardiol.* Vol. 24, pp. 1700–1707, 1994.
- Saeyns Y., Inza I., Larrañaga P., A review of feature selection techniques in bioinformatics, *Bioinformatics*, Vol. 23, No. 19, pp. 2507–17, 2007.
- San O. M., Huynh V.-N., Nakamori Y, A k-Prototypes Algorithm for Clustering Mixed Numeric and Categorical Data, *Sofuto Saiensu. Wakushoppu Koen Ronbunshu*, Vol. 13, pp.75-78, 2003.
- Savaresi M., Boley D.M., On the performance of bisecting k-Means and PDDP, *First SIAM International Conference on Data Mining* , pp 1-14, 2001.
- Scherf U, Ross DT, Waltham M, Smith LH, Lee JK, Tanabe L, Kohn KW, Reinhold WC, Myers TG, Andrews DT, Scudiero DA, Eisen MB, Sausville EA, Pommier Y, Botstein D, Brown PO, Weinstein JN: A gene expression database for the molecular pharmacology of cancer. *Journal of Nature Genetics*, Vol. 66, pp.236-244, 2000.
- Schuckers S., Pisut R., Distinction of arrhythmias with the use of approximate entropy, *Computers in Cardiology*, Hannover, Germany, 1999.
- Schwarz U.I., Ritchie M.D., Bradford Y., Li C., Dudek S.M., Frye-Anderson A., Kim R.B., Roden D.M., Stein C.M., Genetic determinants of response to warfarin during initial anticoagulation, *N Engl J*, Vol. 358, No. 10, pp.999-1008, 2008.
- Sconce E. A., Khan T. I., Wynne H. A., Avery P., Monkhouse L., Barry P., The impact of CYP2C9 and VKORC1 3genetic polymorphism and patient characteristics upon warfarin dose requirements: proposal for a new dosing regimen, *Blood*, Vol. 106, pp. 2329-33, 2005.
- Segal E., Wang H. and Koller D., Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, Vol. 19, pp. i264–i272, 2003.
- Segal E., Yelensky R. and Koller D., Genome-wide discovery of transcriptional modules from dna sequence and gene expression, *Bioinformatics*, Vol. 19, No. 1, 2003
- Shen J, Deng Y, Lee ES, Chang SI, SJ. B, Determination of cluster number in clustering microarray data. *Applied Math and Computation*, Vol. 169, pp. 1172-1185, 2005.
- Sheng Q., Moreau Y., and De Moor B., Biclustering micrarray data by gibbs sampling. In *Bioinformatics*, Vol. 19 (Suppl. 2), pp. 196–205, 2003.
- Sherrah J., Bogner R. E., and Bouzerdoum A., Automatic selection of features for classification using genetic programming, in *Proc. Australian New Zealand Conf. Intelligent Information Systems*, pp. 284-287, 1996.

- Siedlecki W. and Sklansky J., A note on genetic algorithms for large scale feature selection, *Patt. Recognit. Lett.*, Vol. 10, pp. 335-347, 1989.
- Siedlecki W, Sklansky J., On automatic feature selection. *Int. J. Pattern Recogni.* Vol. 2, pp. 197-220, 1998.
- Sinha, S. and Tompa, M., "A statistical method for finding transcription factor binding sites" In *Proceedings International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 344-54, 2000.
- Skalak D., Prototype and feature selection by sampling and random mutation hill climbing algorithms. *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 293-301, 1994.
- Sneath P. H. A. and Sokal R. R., *Numerical Taxonomy*. Freeman, San Francisco, 1971.
- Spellman P.T., Sherlock G., Zhang M.O., Iyer V.R., Anders K., Eisen M.B., Brown P.O., Botstein D. and Futcher B., Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization, *Mol. Biol. Cell*, Vol. 9, No. 12, pp. 3273-3297, 1998.
- Srivastava M. Genomic structure and expression of the human gene encoding cytochrome b561, an integral protein of the chromaffin granule membrane. *J. Biol. Chem.*, Vol. 270, No. 39, pp. 714-720, 1995.
- Stanfill C. and Waltz D. Toward memory based reasoning, *Communication of the ACM* Vol. 29, No. 12, pp. 1213-1228, 1986.
- Steinbach, M., Karypis, G. and Kumar V., A Comparison of Document Clustering Techniques. In *Proc. of KDD Workshop on Text Mining, Proc. of the 6th Intl. Conf. on Knowledge Discovery and Data Mining*, Boston, MA, 2000.
- Strothman R.C. The coming Kuhnian revolution in biology. *Nat. Biotechnol.* Vol. 15, pp. 194-199, 1997.
- Tanay A., Sharan R., and Shamir R.. Discovering statistically significant biclusters in gene expression data. In *Bioinformatics*, vol. 18 (Suppl. 1), pp. S136-S144, 2002.
- Tang C., Zhang L., Zhang I., and Ramanathan M., Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In *Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, pp. 41-48, 2001.
- Taskar B., Segal E., and Koller D., Probabilistic classification and clustering in relational data. In *Proc. of the 17th International Joint Conference on Artificial Intelligence*, pp. 870-878, 2001.
- Tasoulis D.K. and Vrahatis M.N., Unsupervised distributed clustering. In *IASTED International Conference on Parallel and Distributed Computing and Networks*, pp. 347-351. Innsbruck, Austria, 2004.
- Tenenbaum J., and Freeman W., Separating style and content with bilinear models. *Neural Computation*, Vol. 12, pp. 1247-1283, 2000.
- The Gene Ontology Consortium, Gene Ontology: tool for the unification of biology. *Nature Genet.*, 25, 25-29, 2000.
- Vanneschi L., Archetti F., Castelli M. and Giordani I., Classification of Oncologic Data with Genetic Programming, *Journal of Artificial Evolution and Applications*, Vol. 2009, (in press), 2009.
- Varshavsky R, Gottlieb A., Linial M. and Horn D. Novel unsupervised feature filtering of biological data. *Bioinformatics*, Vol. 22, pp. e507-e513, 2006.

- Vaya C., Rieta J., Alcaraz R., Sanchez C., Cervigon R., Prediction of atrial fibrillation termination by approximate entropy in the time–frequency domain, in: *Computers in Cardiology*, Valencia, Spain, pp. 589–592, 2006.
- Voorra D., Eby C., Linder M., Milligan P., Bukaveckas B., McLeod H., Maloney W., Clohisy J., Burnett R., Grosso L., Gatchel S. and Gage B., “Prospective dosing of warfarin based on cytochrome p450 2c9 genotype,” *Thromb Haemost.*, vol. 93, no. 700-705, 2005.
- Wadelius M., Chen L.Y., Eriksson N., Bumpstead S., Ghori J., Wadelius C., Bentley D., McGinnis R., Deloukas P., Association of warfarin dose with genes involved in its action and metabolism, *Hum Genet.*, Vol.121, No. 1, pp. 23-34, 2007.
- Wadelius M., Chen L.Y., Lindh J., Eriksson N., Ghori M., Bumpstead S., Holm L., McGinnis R., Rane A., and Deloukas P., The largest prospective warfarin-treated cohort supports genetic forecasting, *Blood*, vol. 113, no. 4, pp. 784–792, 2009.
- Wagstaff K., Cardie C., Rogers S., Schroedl S., Constrained K-Means clustering with background knowledge. In *Proceedings of 18th International Conference on Machine Learning (ICML-2001)*, pp. 577–584, 2001.
- Weinstein, J.N. et al. An information-intensive approach to the molecular pharmacology of cancer. *Science*, Vol. 275, pp. 343–349, 1997.
- Weka, A multi-task machine learning software developed by Waikato University, 2006. See [www.cs.waikato.ac.nz/ml/weka](http://www.cs.waikato.ac.nz/ml/weka).
- Weston, J., Elisseeff, A., Schölkopf, B., and Tipping, M., Use of the zero norm with linear models and kernel methods. *J. Mach. Learn. Res.*, Vol. 3, pp.1439-1461, 2003.
- Witten H.I. and Frank E., *Data Mining Practical Machine Learning Tools and Techniques with Java Implementation*, Morgan Kaufmann Publishers, San Fransisco, CA, 2000.
- Wolf, L. and Shashua, A. Feature Selection for Unsupervised and Supervised Inference: The Emergence of Sparsity in a Weight-Based Approach, *Journal of Machine Learning Research*, Vol. 6, pp. 1855—1887, 2005.
- Xing E. P., Ng A.Y., Jordan M. I., Russell S., Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, Vol. 15, pp. 505–512, Cambridge, MA, 2003.
- Xu R. and Wunsch D., Survey of clustering algorithms. *Neural Networks, IEEE Transactions*, Vol. 16, No. 3, pp. 645-678, 2005.
- Yu L, Liu H., Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.* Vol. 5, pp. 1205–1224, 2004.
- Zahn C. T. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, Vol. C-20, No. 1, pp. 68–86, 1971.
- Zambelli F., Pesole G., Pavesi G., “Pscan: finding over-represented transcription factor binding site motifs in sequences from co-regulated or co-expressed genes”, *Nucleic Acids Research*, Vol. 37(Web Server issue), pp. W247-W252, 2009
- Zhang B., Hsu B., and Dayal U. K-Harmonic Means—A Data Clustering Algorithm. *Technical Report HPL-1999-124*, Hewlett Packard Laboratories, 1999.
- Zhang Y., Brady M., and Smith S., Hidden Markov random field model and segmentation of brain MR images. *IEEE Transactions on Medical Imaging*, Vol. 20, Num. 1, pp. 45–57, 2001.
- Zhang Z., Gu J. and Gu X., “How much expression divergence after yeast gene duplication could be explained by regulatory motif evolution?” *Trends Genet.* Vol. 20, pp. 403- 407, 2004.

Zhou X.J., Kao M.C., Huang H., Wong A., Nunez-Iglesias J., Primig M., Aparicio O.M., Finch C.E., Morgan T.E. and Wong W.H., "Functional annotation and network reconstruction through cross-platform integration of microarray data", *Nat. Biotechnol.*, Vol. 23, pp. 238-243, 2005.

Zweig M. H. and Campbell G., "Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine," *Clinical Chemistry*, Vol. 39, No. 4, pp. 561-577, 1993.