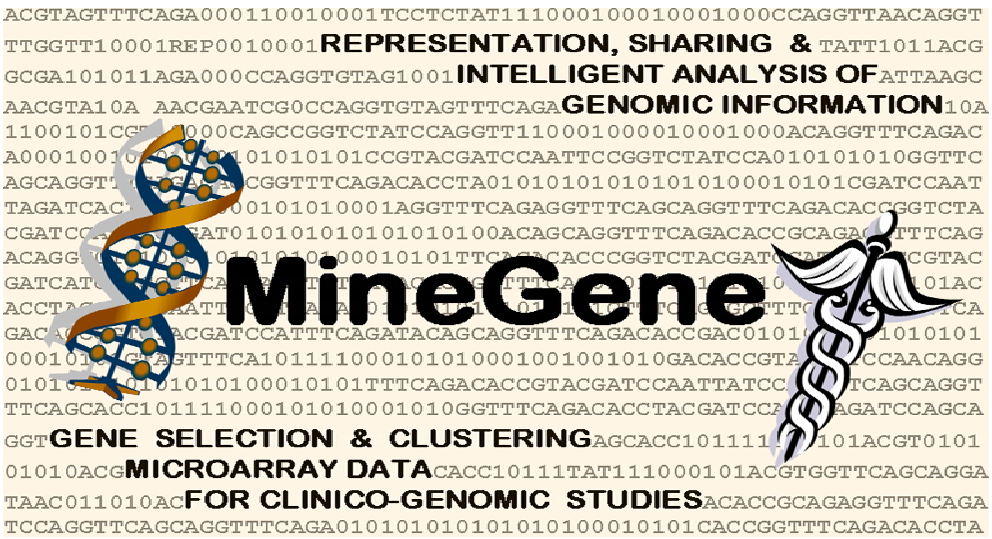
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ



Thesis Title

Gene Selection & Clustering Microarray Data: The MineGene System

Αλέξανδρος Καντεράκης

Ηράκλειο, Κρήτη
Απρίλιος 2005

Gene Selection & Clustering Microarray Data: The MineGene System

Kanterakis Alexandros
Master of Science

Department of Computer Science
University of Crete

Abstract

Over the last years we witness a revolution initiated by the completion of the Human Genome Project. DNA, the molecule that encodes our genetic information, has been fully sequenced setting new promises and challenges for understanding the role of genetic factors in human health and diseases. Moreover, DNA Microarrays are devices that measure the expression of many thousands of genes in parallel permitting the rapid profiling of gene expressions. Although these technological advances lead us to the understanding of the genetic base of various diseases it is evident that we need to integrate the knowledge normally processed in the clinical setting. In this Thesis we present firstly the features and components of a seamless modern information system for microarray data management that follows specific well-known ontologies and annotations alongside with some existing implementations. Furthermore we envisage a synergic clinico-genomic decision making scenario, where patient's genotypic and phenotypic profile will be utilized for disease diagnose and treatment. Consequently we present two novel machine learning algorithms that facilitate the integration of such data in the medical decision process. The first is a supervised gene selection algorithm based on gene ranking through an entropic metric. The second is an unsupervised graph theoretical hierarchical clustering approach. These methods have been implemented and applied to real-world datasets and compared to other published approaches.

Supervisor: Plexousakis Dimitris
Associate Professor

Επιλογή Γονιδίων και Κατάτμηση Δεδομένων Πειραμάτων με Μικροσυστοιχίες: Το σύστημα MineGene

Καντεράκης Αλέξανδρος
Μεταπτυχιακή εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Τα τελευταία χρόνια είμαστε μάρτυρες μίας επανάστασης η οποία ξεκίνησε με την ολοκλήρωση της αποκωδικοποίησης του ανθρώπινου γονιδιώματος. Το DNA, το μόριο που περιέχει τις γενετικές μας πληροφορίες έχει αναλυθεί θέτοντας νέες υποσχέσεις και προκλήσεις για την κατανόηση του ρόλου των γενετικών παραγόντων στην υγεία και ασθένεια των ανθρώπων. Επιπλέον, οι μικροσυστοιχίες DNA είναι συσκευές που μετράνε τη ταυτόχρονη έκφραση πολλών χιλιάδων γονιδίων, επιτρέποντας την ταχύτατη καταγραφή της έκφρασης των γονιδίων. Παρόλο που αυτές οι τεχνολογικές εξελίξεις μας οδηγούν στην κατανόηση της γενετικής βάσης διάφορων ασθενειών είναι προφανές ότι πρέπει να συμπεριλάβουμε και την γνώση που έχει αποκτηθεί από την κλινική εμπειρία. Στην παρούσα εργασία παρουσιάζουμε αρχικά τις ιδιότητες και τα κύρια στοιχεία ενός αποκεντρωμένου σύγχρονου πληροφοριακού συστήματος το οποίο χρησιμοποιεί συγκεκριμένες γνωστές οντολογίες και επισημειώσεις για διαχείριση δεδομένων που έχουν παραχθεί από πειράματα με μικροσυστοιχίες και αναλύουμε διάφορες υπάρχουσες υλοποιήσεις. Στη συνέχεια οραματιζόμαστε ένα συνεργικό κλινικό-γενομικό σενάριο λήψης αποφάσεων όπου ο γονότυπος και ο φαινότυπος των ασθενών θα αξιοποιείται για διάγνωση και θεραπεία. Επιπροσθέτως παρουσιάζουμε δύο πρωτότυπους αλγόριθμους μηχανικής μάθησης που υλοποιούν την ολοκλήρωση αυτών των δεδομένων κατά τη διαδικασία λήψης ιατρικών αποφάσεων. Ο πρώτος είναι ένας επιβλεπόμενος αλγόριθμος για επιλογή γονιδίων ο οποίος βασίζεται στη βαθμολόγηση γονιδίων μέσω μίας εντροπικής μετρικής. Ο δεύτερος είναι ένας μη επιβλεπόμενος γραφο-θεωρητικός αλγόριθμος για ιεραρχική κατάτμηση. Τέλος παρουσιάζουμε την υλοποίηση των παραπάνω μεθόδων, την εφαρμογή τους σε πραγματικά δεδομένα καθώς επίσης και την σύγκρισή τους με άλλες δημοσιευμένες προσεγγίσεις.

Επόπτης: Πλεξουσάκης Δημήτριος
Αναπληρωτής Καθηγητής

Ευχαριστίες

Όταν ξεκίνησα την μεταπτυχιακή μου εργασία πριν από 2 χρόνια περίπου δεν μπορούσα να φανταστώ ότι η εφαρμογή υπολογιστικών μεθόδων στη βιολογία μπορεί να είναι μία τόσο συναρπαστική εμπειρία. Σήμερα, με την εργασία ολοκληρωμένη, νοιώθω την ανάγκη να ευχαριστήσω καταρχήν τον επόπτη καθηγητή μου Γιώργο Ποταμιά όχι μόνο για την συνεχή και ουσιαστική καθοδήγηση που μου παρείχε, αλλά και για τη γνωριμία με έναν τρόπο σκέψης και εργασίας με κύρια συστατικά την συνεχή επαφή με τις εξελίξεις, την εφαρμογή πρωτότυπων ιδεών και την αρμονική και δημιουργική συνεργασία.

Επίσης θέλω να ευχαριστήσω την Ερευνήτρια του ΙΤΕ-ΙΠ Αναστασία Αναλυτή, και τους μεταπτυχιακούς φοιτητές Χάρη Κονδυλάκη και Δημήτρη Μανακανάτα για την βοήθεια τους στην επιλογή και διαμόρφωση της γενομικής βάσης δεδομένων. Επίσης, ο Θανάσης Μαργαρίτης έλυσε πολλές απορίες που είχαν να κάνουν με βιολογία και γενετική, ο Μανόλης Σπανάκης έδωσε πολύτιμες συμβουλές σε θέματα προγραμματισμού και ο Δημήτρης Γάκης διόρθωσε αρκετά γραμματικά λάθη που περιείχε η αναφορά.

Ιδιαίτερα ευχαριστώ επίσης τον ακαδημαϊκό μου σύμβουλο, αείμνηστο Στέλιο Ορφανουδάκη και τον αντικαταστάτη του καθηγητή Δημήτρη Πλεξουσάκη. Επίσης τον καθηγητή Γιάννη Τόλλη για την συμμετοχή του στην εξεταστική επιτροπή αλλά και για τις πολύτιμες συμβουλές που μου έδωσε.

Τέλος, θέλω να ευχαριστήσω μία παρέα δραστήρια, τολμηρή, γεμάτη προκλήσεις και ομορφιά με ετερόκλητους τόπους διαμονής, θέσεις και ιδέες. Αποτέλεσαν κάτι παραπάνω από μια ευχάριστη παρέα και μου έδωσαν περισσότερα από συμπαράσταση και ψυχολογική υποστήριξη. Ιδιαίτερα ευχαριστώ την φίλη μου Δέσποινα Αντωνακάκη και την οικογένεια μου, την αδερφή μου Βασιλική και τους γονείς μου Νίκο και Ελένη στους οποίους και αφιερώνεται αυτή η εργασία.

CONTENTS

List of Figures

List of Tables

1. Introduction

The completion of a high-quality, comprehensive sequence of the human genome [1], is a landmark event commencing the genomic era. Genome sequences, the bounded sets of information that guide biological development and function, lie at the heart of this revolution making genomics a central and cohesive discipline of biomedical research [2]. Our ability to explore genome function is increasing in specificity as each subsequent genome is sequenced. The practical consequences of the emergence of this new field are widely apparent. Identification of the genes responsible for human diseases, once an effortful task requiring large research teams, many years of hard work, and an uncertain outcome, can now be routinely accomplished in a reasonable time space by a single specialist with access to DNA samples and associated phenotypes, an Internet connection to the public genome databases and a DNA-sequencing machine or microarray device.

Microarray technology provided the ability to explore gene expression of tens of thousands of genes in a time-feasible scale [3]. They are mainly used to estimate differential expression of genes acquired from tissues in various states and conditions, making practical comparisons between a sample genotype profile and an arbitrary phenotype attribute or clinical observation. This linkage promise to bring close to reality one of the most ambitious visions of modern medicine: The embodiment and unification of clinical and genomic medicine. The sequencing of the human genome, along with other recent and expected achievements in genomics, provides an unparalleled opportunity to advance our understanding of the role of genetic factors in human health and disease, to allow more precise definition of the non-genetic factors involved, and to apply this insight rapidly to the prevention, diagnosis and treatment of disease [4], [5]. Thus, clinical opportunities for individualized gene-based pre-symptomatic prediction of illness and adverse of drug response are emerging at a rapid pace [6].

Although genome-based analysis methods are rapidly permeating biomedical research [7], the challenge of establishing robust paths from genomic information to improved human health remains immense. In the field of microarray experiments in particular a wide range of computational requirements have arisen, including image processing [8], instrumentation and robotics [9], database design [10], [11], data storage and retrieval [12], microarray design based on available Expressed Sequence Tags (ESTs) [13], and data analysis [14]. Furthermore, microarray data need to be interpreted in the context of other biology knowledge, involving various types of post-genomics informatics [15], including gene networks [16], gene pathways [17], and gene ontologies [18].

In this thesis we present a novel approach for microarray gene expression data management and analysis. We introduce a seamless environment where gene expression data are submitted, stored, queried, retrieved, visualized and shared among researchers inside and outside the laboratory space. Each database transaction follows well-known and accepted data standards, ontologies and annotations. This environment is enriched with a plug-in software environment able to perform supervised and unsupervised machine learning algorithm in order to extract invaluable information about the inherent gene regulations. The implemented algorithms include some well-known learning algorithms like Support Vectors Machines (SVM [19]) and K-Means [20] as well as some novel approaches based on

the application of an entropic metric for gene discretisation for supervised learning and the hierarchical clustering of a Minimum Spanning Tree for a Graph Theoretic Clustering approach. Our methods have been applied on various real-world gene expression domain studies and their superiority has been shown.

Moreover, we present a synergistic clinico-genomic decision-making scenario where through microarray gene expression profiling we will be able to link potential phenotypical profiles to respective molecular or, genotypical ones. Such advancement may be utilized in the course of both prognostic and therapeutic decision-making processes.

1.1 DNA Microarrays

DNA Microarrays are devices than can estimate in parallel, the expression of many thousands of genes. Their invention in 1995 [21] brought a revolution in molecular biology, and in the past six years their use has grown rapidly in medicine as well as in pharmaceutical, biotechnology and food industries [3].

Microarray technology makes use of the sequence resources created by the genome projects such as the Human Genome Project [22], the Dog Genome Project [23] and the Mouse Genome Sequencing Consortium [24] as well as other sequencing efforts. The main question microarrays are posed to answer is what genes are expressed in a particular cell type of an organism, at a particular time, under particular conditions. For instance, they allow comparison of gene expression between normal and diseased (e.g. cancerous) cells.

A DNA microarray consists of a solid surface, usually a microscopy slide, onto which DNA molecules have been chemically bonded at fixed locations, called spots. There may be tens of thousands of spots on an array, each containing a huge number of identical DNA molecules, of lengths from twenty to hundreds of nucleotides [25]. For gene expression studies, each of these molecules should ideally identify on gene or one exon in the genome, however in practice this is not always so simple and may not even be possible due to families of similar genes in genome. Microarrays that contain all of the about 6000 genes of the yeast genome have been available since 1997 [26]. The spots are either printed on the microarrays by a robot, or synthesized by photo-lithography (similarly as in computer chip production) or by ink-jet printing.

Although microarrays are used in many research interests such as the identification and location of SNPs (Single Nucleotide Polymorphism), the major microarray application is to detect the presence and abundance of labeled nucleic acids in a biological sample, which will hybridize to the DNA on the array via Watson-Crick duplex formation [27] and which can be detected via the label. In the majority of microarray experiments, the labeled nucleic acids are derived from the mRNA of a sample or tissue, and so the microarray measures gene expression. The power of microarray is that there may be many thousands of different DNA molecules bonded to an array, and so it is possible to measure the expression of many thousands of genes simultaneously [28].

There are different ways in which microarrays can be used to measure the gene expression levels. One of the most popular microarray applications allows us to compare gene expression levels in two different samples, e.g. the same cell type in a healthy and diseased state. The total mRNA from the cells in two different conditions is extracted and labeled with two different fluorescent labels: for example a green dye

for cells at condition 1 and a red dye for cells at condition 2. To be more accurate, the labeling is typically done by synthesizing single stranded DNAs that are complementary to the extracted mRNA by an enzyme called reverse transcriptase. Both extracts are washed over the microarray. Labeled gene products from the extracts hybridize to their complementary sequences in the spot due to the *preferential binding*, as it is called the tendency of the complementary single stranded nucleic acid sequences to attract each other. Moreover the longer the complementary parts, the stronger the attraction. Hybridization is the major process of a microarray experiment [29]. Two DNA strands hybridize if they are mutually complementary, that is when adenine (A) binds with thymine (T) and cytosine (C) binds with guanine with Watson-Crick hydrogen bounds (figure 1).

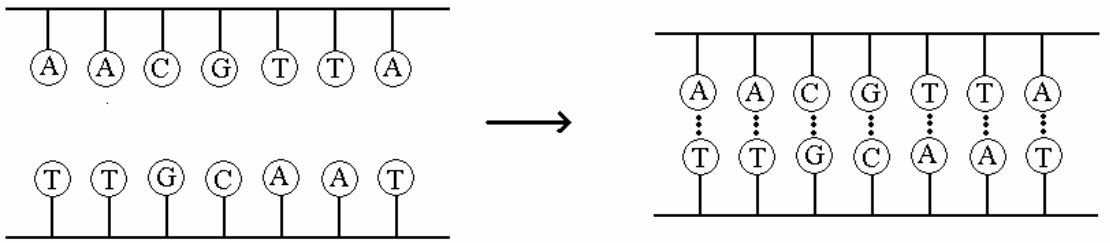


Figure 1. Hybridization of two DNA molecules. The dashed lines show the hydrogen bounds.

The dyes enable the amount of sample bound to a spot to be measured be the level of fluorescence emitted when it is excited by a laser. If the RNA from the sample in condition 1 is in abundance, the spot will be green, if the RNA from the sample in condition 2 is in abundance, it will be red. If both are equal the spot will be yellow, while if neither is present it will not fluoresce and appear black. Thus, from the fluorescence intensities and colors for each spot, the relative expression levels of the genes in both samples can be estimated. The raw data that are produced from microarray experiments are the hybridized microarray images. To obtain information about gene expression levels, these images should be analyzed, each spot on the array identified, its intensity measured and compared to the background. This is called image quantitation (figure 2).

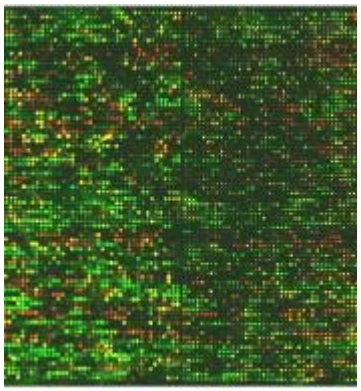


Figure 2. An illuminated microarray (enlarged). A typical size of such an array is about 1.5 cm or less. The diameter of each spot is of the order of 0.1nm, for some microarray types can be even smaller [2]**.**

Image quantitation is done by image analysis software. To obtain the final gene expression matrix from spot quantitation, all the quantities related to some gene (either on the same array or on arrays measuring the same conditions in repeated experiments) have to be combined and the entire matrix has to be scaled to make different arrays comparable. This process is called normalization. The final extract is a numerical matrix containing the relative expression of each gene in the healthy cell versus the diseased cell.

One of the principal features of microarrays is the volume of quantitative data that they generate. As a result, the major challenge is how to handle, interpret and make use of this data. The field of bioinformatics promises to deal with this aimed by the applications of mathematics, statistics and information technology.

1.1.1 DNA Microarray Technologies

In microarray experiments, each spot contains either DNA oligomers, or a longer DNA sequence designed to be complementary to a particular mRNA of interest. The choice of spotting oligomers or a longer cDNA sequence yields two different microarray technologies: oligo and cDNA microarrays respectively. Oligo arrays are generated by photolithography techniques to synthesize oligomers directly on the glass slide. These arrays are manufactured and marketed primarily by Affymetrix Inc [30]. In contrast, cDNA arrays are created by mechanical gridding, where prepared material is applied to each spot by ink-jet or physical deposition.

There is generally a one-to-one correspondence between spots and genes, but various exceptions hold. Multiple genes may hybridize to the same spot if the DNA at that spot is not unique to a single gene. This problem is called cross-hybridization. Likewise, a gene may hybridize to more than one on a microarray if different spots cover different regions of the gene. In fact, many microarrays are designed deliberately to identify individual exons of a gene, in order to study expression patterns for different splice forms or transcripts. Because of these considerations, it is more accurate to say that each spot measures one or more transcripts of a gene, rather than a particular gene. In this thesis we will consider that we are measuring expression levels of genes rather than transcripts.

Because cDNA sequences on a microarray are hundreds of nucleotides long, a single spot is usually sufficient to identify a particular gene. However, oligo microarrays have spots that contain oligomers of 25 or so nucleotides. Because such short oligomers will frequently cross-hybridize with several genes, oligo arrays must measure each gene with several oligomers (16-20 in the Affymetrix arrays). Each set of oligomers is called a probe set. A gene is considered present only when the vast majority of the probe set shows positive hybridization.

Oligo microarrays also have another special feature, designed to account for the fact that short oligomers can have non-specific binding and can vary in their hybridization efficiency. Each oligomer on the array has a mismatch oligomer which is intended to serve as a control. The mismatch oligomer is the same as its corresponding perfect match oligomer except for one position, which is designed to be different. The amount of specific hybridization can then be measured by taking the difference in hybridization between the perfect match and its corresponding mismatch.

More than one sample may be applied to a single microarray, with the different samples being labeled with differently colored dyes. In practice, however, Affymetrix oligo arrays measure a single sample at a time and therefore use a single type of dye. In contrast, cDNA microarrays measure either on sample or, more commonly, two samples.

1.2 Gene-Expression Data Analysis

Microarray data analysis is heavily depended on Gene Expression Data Mining (GEDM) technology, and in the very-last years a lot of research efforts are in progress. GEDM is used to identify intrinsic patterns and relationships in gene expression data. The identification of patterns in complex gene expression datasets provides two benefits:

- Generation of insight into gene transcription conditions
- Characterization of multiple gene expression profiles in complex biological processes, e.g. pathological states

GEDM activities are based on two approaches:

- *Hypothesis testing*: to investigate the induction or perturbation of a biological process that leads to predicted results – in this case the basic task is to identify *gene-markers* or, *molecular signatures* of a disease or a disease state, and
- *Discover hidden regularities*: to detect internal structure in biological data – in this case the basic task is to uncover *hidden regularities* in gene-expression data, an *exploratory* data analysis to find *genes of similar profiles* (across patient samples) and (potentially) *co-regulated*.

In this context advanced data mining technologies- clustering, classification, and visualization are utilized. The final goal is the delivery of an operational *Gene Expression Data Mining Suite* (GEDMS) to accommodate a set of smoothly integratable data-mining tools. The aim is to help the clinicians and molecular biologists in their research and data processing enquires.

1.2.1 Intelligent Processing: From Internal Data Regularities to Gene Markers

By measuring transcription levels of genes in an organism under various conditions, in different tissues, we can build up 'gene expression profiles', which characterize the dynamic functioning of each gene in the genome. The microarray data are represented in a matrix with rows representing genes, columns representing samples (e.g. various tissues, developmental stages and treatments), and each cell containing a number characterizing the expression level of the particular gene in the particular sample, i.e., the gene expression matrix.

There are two straightforward ways how gene expression matrix can be studied:

- Comparing expression profiles of genes by comparing rows in the expression matrix.
- Comparing expression profiles of samples by comparing columns in the matrix.

Additionally, both methods can be combined (provided that the data normalization allows it).
When comparing rows or columns, we can look either for similarities or for differences and accordingly form *classification*-rules and *clusters* [31]. Clustering and

classification results may reveal *correlations* between expression of certain genes and guide to the identification of:

- disease occurrence or state (malignant vs. healthy vs. benign tissue),
- disease clinical markers such as tumor type, stage, size etc,
- disease prognosis or treatment outcome, and
- provide information about the genetic profile of different subgroups of disease-types (e.g., breast cancer) and possibly identify new subgroups or merge together subgroups that were previously believed to be genetically separate.

1.2.1.1 Difficulties of Gene Expression data analysis

Although microarray experiments are a breakthrough in molecular biology and genomics, the whole experimental procedure is very complex and error-prone. Raw data produced by microarray experiments are subject to errors caused by many factors that most of them are still open problems [32].

First of all, like many experimental technologies, microarrays measure the target quantity (i.e. relative or absolute mRNA abundance) indirectly by measuring another physical quantity, the intensity of the fluorescence of the spots on the array for each fluorescent dye, i.e. for each optical wavelength (so-called channel). Therefore the raw data produced by microarrays are in fact images (figure 2). Transforming these images into the gene expression matrix is a non-trivial process: the spots corresponding to genes on the microarray should be identified, their boundaries determined, the fluorescent intensity from each spot measured and compared to the background intensity and to these intensities for other channels. The software for this initial image processing is often provided with the image scanner, since it will depend on particular properties of the hardware. A survey of image analysis software can be found at [33].

In any physical experiment it is important to know not only the value of the measurement, but also the standard error or some other indicator of reliability for each data point. For most microarray technology platforms only the ratio of the background-subtracted signals of the given sample and the control is meaningful. If the spot intensity is low, the ratio of these numbers may be high, but the measurement may not be reliable. The spot quality can be assessed not only by the absolute intensity in each channel, but also by many other factors, such as uniformity of the individual pixel intensities, or the shape of the spot. Unfortunately there is currently no standard way of assessing the spot measurement reliability. If experiments have been done in replicates, they can be used to assess the standard error in addition to the single measurement quality assessments. Little has been published yet in how to use the reliability of gene expression measurements by combining the information about the spot image in each channel and the replicate images.

Another difficulty in creating a gene expression matrix comes from the necessity to identify each spot with the respective gene. This is not always possible, since spots are typically based on EST sequences. EST (Expressed Sequence Tag) is usually short single read from mRNA (cDNA) which is usually produced in large numbers. It represents a snapshot of what is expressed in a given tissue, and/or at a given developmental stage. It also represents tags (some coding, others not) of expression for a given CDNA library [34]. Linking EST to the respective gene may be non-trivial. Typically it is done through EST clustering. Additionally, the same gene may be

represented by several spots on the array, rather by exactly the same or by different sequence. The problem arises when we measure different expression levels from these spots.

Microarray-based gene expression measurements are still far from giving estimates of mRNA counts per cell in the sample. The measurements are relative by nature: essentially we can compare the expression level either of the same gene in different samples, or of different genes in the same sample. Moreover, appropriate normalization should be applied to enable any data comparisons. Typically it is assumed that abundance ratios of 1.5-2.0 are indicative of a change in gene expressions but such estimates are very crude. The reliability of ratios depends on the absolute intensity values, as well as varying from spot to spot due to specificity of the sequence and cross-hybridization of homologous sequences. This should be kept in mind while analyzing the gene expression matrix. The value of microarray-based gene expression measurements would be considerably higher if reliability and limitations of particular microarray platforms for particular kinds of measurements, as well as cross-platforms comparison and normalization were studied and published.

An outline of the overall microarrays' experimental methodology and the resulting gene-expression matrix is illustrated in figure 3 (below).

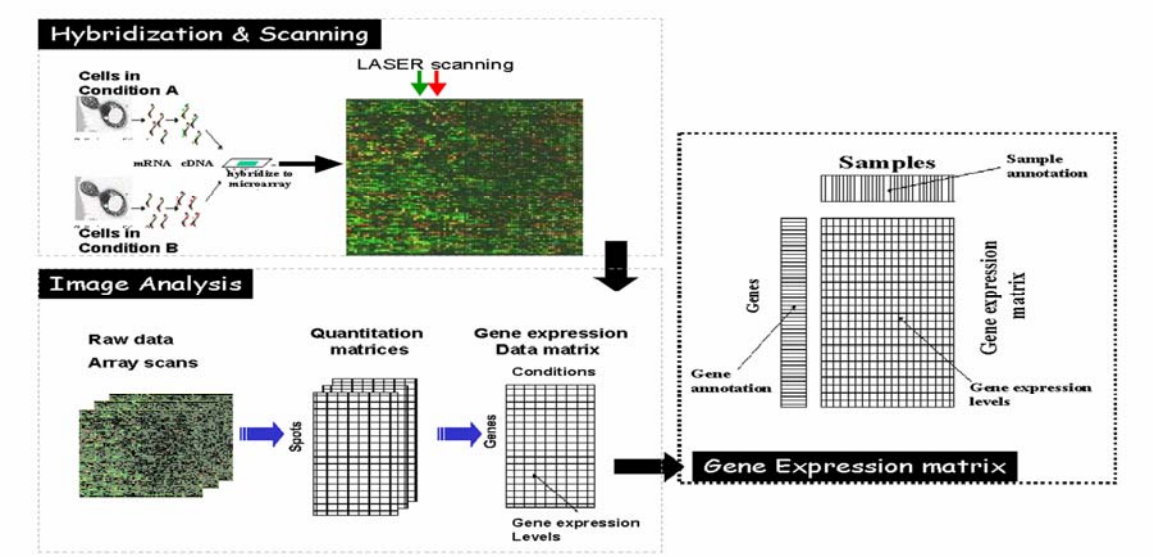


Figure 3. Microarrays: *Experimental set-up and resulting gene-expression matrix.*

1.3 Classification and Gene Expression Profiling

Classification is a supervised intelligent data analysis approach. One of the goals of supervised expression data analysis is to construct classifiers, such as decision trees or, rules; support vector machines (SVM), or, trained Artificial Neural Networks (ANN), which assign predefined classes to a given expression profile. For instance, if a classifier can be constructed based on a gene expression profile that is able to distinguish between two different, but morphologically closely related tumor tissues, such a classifier can be used for diagnostics. Moreover, if such a classifier is based on a set of relatively simple rules, it can help to understand what are the mechanisms involved in various tumor samples.

Typically, such classifiers are trained on a subset of data with a-priori assigned classification (e.g., tumor-type-X vs. tumor-type-Y samples). The outcome (decision tree/rules, SVM or, trained ANN) is then tested and evaluated on another subset with known classification. After assessing the quality of the prediction they can be applied to data the classification of which is unknown. In this mode, the classifier could be utilized in order to classify new incoming data of unknown class, e.g., to predict the tumor-type of sample-tissues based on their respective expression-profile.

By comparing samples, we can find classification-archetypes (class descriptions) with which differentially expressed genes are combined to distinguish between the samples (e.g., normal vs. cancerous samples). So, we may be able to identify *'discriminant'* genes that relate to the clinical-profiles of specific patient-groups, and to study the effect of various chemotherapeutic-treatments. Such a data-analysis scenario composes a more targeted-research line of work, aiming towards the device of diagnostic or and/or prognostic tests.

1.3.1 Discriminatory Gene Selection

The problem now is how to select the genes that best discriminate between the different disease states. The problem is well-known in the machine learning community as the problem of feature-selection (with its dual 'feature-elimination'), and various 'wrapper-based', or, 'filtering' approaches have been proposed.

Traditionally, in machine learning research the number of features, m, is quite smaller than the number, k, of cases (samples in the case of gene-expression studies) that is, $m << k$. In contrast, gene-expression studies refer to a huge number of features and quite few samples. In most gene-expression domains the number of genes is in the range of 2000 – 35000 (i.e., the estimated number of human genes), and the number of samples in the range of 50 – 200, that is $k << m$. In a situation like that it is questionable if a 'wrapper' based feature-selection approach could help, because of its high-computational cost. So, in most gene-selection studies a 'filtering' approach is followed.

The gene-selection methods are used in order to estimate the *correlation strength* of genes that appear in important clusters with any of the samples' categories (i.e., disease-types; disease-recurrence states etc). Genes with *high ranked ordered correlation scores* will be proposed as possible indicative markers for these categories. Special attention will be paid to genes whose expression changes very slightly in malignant tissue, as these may represent genes activated in initial phases of the disease and may provide insight about the biological origin of a disease.

- *The utility of discriminating genes in prognosis.* When trying to predict treatment outcome, selected genes will be used to identify which patients will respond well to treatment and which not. To predict disease recurrence, selected genes are used to identify patients that will be disease-free after a certain time period and those at high risk of developing metastases. This type of categorization problem is a supervised classification task where a priori information about the correct categorization of a group of patients is used to teach the method to learn the intrinsic relationships between the selected gene expressions of patients within each category. This relationship is often very complex and cannot be described by traditional similarity or distance metrics as the ones used in clustering methods. Once the classification tool learns these relationships, it can be used to predict the probability of clinical categories (e.g. metastatic or cancer-free) for

new patients by looking at their respective expression profile over the selected gene set. Such methods can be used to automatically predict probabilities for the expected treatment outcome or disease recurrence for new patients.

Prediction performance of gene-selection (as well classification) methods should be tested in full cross-validated settings and additionally with new patient data as they accumulate over time to ensure that predictions are reliable over a broad range of breast cancer types. As more samples accumulate, re-selection of suitable gene markers might be necessary to ensure that information is learnt for as many disease-types as possible. Re-teaching the classifiers will further enhance their prediction performance.

1.3.2 Clustering and Gene Expression Profiling

The goal of clustering is to group together objects (genes or samples) with similar properties. This can also be viewed as the reduction of the dimensionality of the system or, the discovery of "structure in the data". Clustering is not a new technique, many algorithms have been developed for it and many of these algorithms have been applied to analyze expression data.

With gene expression data analysis we try to identify the changing and unchanging levels of gene expression and to correlate these changes to identify sets of genes with similar profiles. The assumption behind- and the utility-of clustering gene-expression data is that 'genes with similar profiles, i.e., in the same cluster, are also co-regulated'. So, clustering may give rise to valuable information about the molecular status of various genes and their functioning. In some cases a mainly visual analysis has been successful in grouping genes into functionally relevant classes. In other studies, simple sorting of expression ratios and some form of 'correlation distance' were used to identify similar genes.

The literature on statistical clustering is fairly vast, offering many other choices of clustering methods. The hierarchical, and k-mean clustering algorithms as well as self-organizing maps have all been used for clustering expression profiles. By comparing gene-expression profiles, and forming clusters, we can hypothesize that the respective genes are co-regulated and possibly functionally related. Then, we may go back to the respective genes DNA-sequences to identify putative transcription-factors (or even identify SNPs- single nucleotide polymorphisms). Such a data-analysis scenario composes a more basic-research line of work.

At the moment, there do not seem to exist any objective guidelines regarding the choice of a clustering algorithm to be used for grouping genes based on their expression profiles. For indicative references about microarrays and gene expression profiling methodologies as related to classification, gene-selection and clustering you may look at references [35]-[72].

- **The current thesis tackles and presents real *innovative approaches*, *algorithms* and *tools* for gene-selection and clustering of gene-expression data – the introduced methods are extensively tested on real-world domains and datasets.**
- **Moreover, a specific objective fullfilled by the current thesis was to *review* the state-of-the-art approaches, methods and tools for the uniform *modelling*, *representation* and *seamless sharing* of the involved *biomedical information* and *data* (i.e., microarray experiments information and gene expression data).**

1.4 Organization of the Thesis

- In chapter 2 we present the existing status in genomic informatics. Specialized databases for sequence storage, genome management, and gene expression are analyzed. Moreover we focus on data standards for microarray experiment activities and we concentrate on standards maintained by the most appreciated consortium in gene expression research area; the MGED group. Furthermore, we present the most acknowledged ontologies for taxonomies, genes and microarrays. We finally perform a comparison between the two most appreciated microarray expression databases ArrayExpress and Base and we highlight the pros and cons of each database system.
- In chapter 3 we focus on the 'old genomics' their limited implications in healthcare and the advances of the 'new genomics' where clinical observations and knowledge coming from gene expression profiling can be integrated into a qualitative predictive and therapeutic healthcare system. We discuss the major applications of genomic medicine in healthcare and the necessity of clinical and genomic integration. We finally present a scenario where phenotypical profiles are linked with genotypical to provide a prognostic or therapeutic decision-making process.
- In chapter 4 we justify the general concept of supervised gene expression database mining, research pathway and the related work. Then we propose a novel gene selection methodology based on gene discretisation and composed by four main modules: gene ranking, gene grouping, consecutive feature elimination and class prediction. Furthermore, we apply the algorithm in real-world datasets and we perform a comparison survey based on the resulted accuracy and feature elimination of our method versus other related methods.
- In chapter 5 we introduce a novel Graph Theoretic Clustering algorithm based on the hierarchical clustering of a Minimum Spanning Tree. This algorithm has the special feature to combine different information sources, in order to estimate the distances between genes, and in order to estimate a special category utility that determines whether the clustering will proceed or not. Then the time complexity is estimated and a heuristic for feasible distance calculation is presented.
- In chapter 6 we present an implementation of all aforementioned methods, algorithms and heuristics. The software system presented is planned to act as a plug-in in microarray gene expression databases, in order to act as a general machine learning algorithm toolkit. It is designed according the principles of object oriented programming it is component based and expandable. The general parameters, inputs, outputs and usage are finally presented.
- Finally, in chapter 7 we conclude the major findings and we describe possible future work.

2. From Genomic to Post-Genomic Informatics: Modeling, Representing and Sharing Genomic Data

Microarrays are already producing massive amounts of data. These data, like genome sequence data, can help us to gain insights into underlying biological processes only if they are carefully recorded and stored in databases, where they can be queried, compared and analyzed by different computer programs.

In this chapter we look at the sequence databases that are used to select and annotate the genes that the microarray detects. Databases are separated in sequence storage, genome management, and gene expression storage. Moreover we focus on data standards for microarray experiment activities and we concentrate on standards maintained by the most appreciated consortium in gene expression research area; the MGED group. Furthermore, we present the most acknowledged ontologies for taxonomies, genes and microarrays. We finally perform a comparison between the two most appreciated microarray expression databases ArrayExpress and Base and we highlight the pros and cons of each database system.

2.1 Sequence Databases

Worldwide there are three major sequence databases:

- The EMBL Nucleotide Sequence Database [73], [74] constitutes Europe's primary nucleotide sequence resource. Main sources for DNA and RNA sequences are direct submissions from individual researchers, genome sequencing projects and patent applications.
- Genbank [75] is the National Institute of Health (NIH) [76] molecular database which is composed of an annotated collection of all publicly available DNA sequences [77]. The February 2004 release of the Genbank molecular database contained 32,549,400 DNA sequences which are further composed of approximately 37,893,844,733 deoxyribonucleotides [78].
- DDBJ (DNA Data Bank of Japan [79]) began DNA data bank activities in 1986 at the National Institute of Genetics (NIG) of Japan.

They date back to 1982, when it became clear that there was a need to publish and share DNA sequences. The American initiative, GenBank and the European initiative, EMBL (European Molecular Biology Laboratory), were launched simultaneously in June 1982, each with approximately 600 sequences. Since that time, the sizes of the databases have grown exponentially, doubling approximately every 17 months (figures 4 and 5). In 1987, the DDBJ was started as a Japanese equivalent of GenBank and EMBL. In 1992, the three databases entered in a collaboration to share all sequences. Since that date, the three databases contain almost identical sequence information. Any sequence submitted to one of the databases will automatically be added to the other two. They also hold two meetings, the International DNA Data Banks Advisory Meeting and the International DNA Data Banks Collaborative Meeting.

The success of these sequence databases has resulted not only from the advances in sequencing technology, but also from the advances in computer technology. The databases require significant computing power and storage to operate, but most important is the role of the Internet. Being online, they offer the ability to anyone to submit, query and download sequences acting as an invaluable medium between research laboratories.

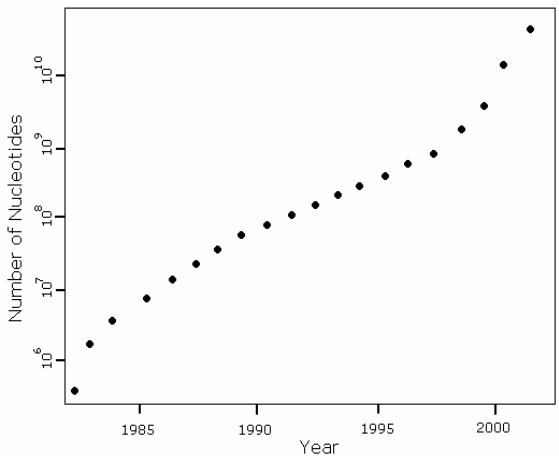


Figure 4. Log plot of the number of nucleotides in the GenBank database between 1982 and 2001. The straight line is indicative of exponential growth in the number of nucleotides, with a doubling time of approximately 17 months.

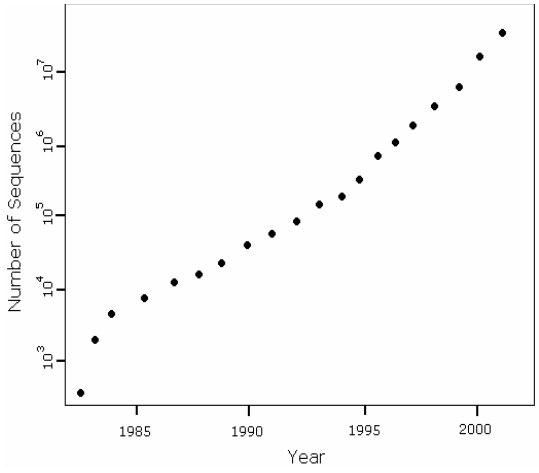


Figure 5. *Log plot of the number of sequences in GenBank database between 1982 and 2001. There is similar exponential growth as with nucleotides.*

- *Problems and Limitations of Sequence Databases.* Although sequence database is the first place to visit when querying a sequence in order to obtain information about it, there are two reasons why they are not sufficient for microarray design and annotation. First, they do not contain meta-information. Although we can identify a sequence taking part in a microarray experiment, we cannot identify the gene from which the EST was derived. Second, sequence databases contain too many sequences for array design. When designing an array, we would want the database to be able to provide a list of genes in which each gene that will appear on the array will appear once in the list. There are two reasons why primary sequence databases cannot provide this. First of all each gene can be represented several times in the database, for example, if it were submitted by different research groups who have sequenced it. Secondly each gene sequence may be in the database in several forms, e.g. as a gene sequence, genomic sequence and as ESTs. The first problem is referred as the redundancy problem and the second as the replication problem. Although sequence databases are used for annotating sequences that appear on arrays, they are not used for array design. This is a domain for secondary sequence databases specialized for microarray experiments.

2.1.1 Secondary Sequence Databases

The three secondary sequence databases that are commonly used for microarray design are UniGene, TIGR Gene Indices and RefSeq.

- UniGene [80] is the database with the greatest historical use for selecting sequences for microarrays. It is an attempt to partition GenBank sequences into clusters, each of which is intended to represent a unique gene. The clusters themselves may contain both mRNA sequences and ESTs, so that they represent both known genes and putative genes based on expressed material that has been sequenced. The clusters are built by comparing all mRNA and EST sequences in GenBank and assigning overlapping sequences to the same cluster [81]. In clusters that contain full-length mRNAs, the task is straightforward, because all ESTs deriving from the gene will align with the same mRNAs. However, many clusters in UniGene contain only ESTs; the algorithms by which UniGene is built assemble the clusters out of overlapping ESTs in order to produce a picture of the gene from which the ESTs have putatively derived. UniGene is available for a range of species. Although there are 50 species in the database, there is only broad coverage of the main research species. The human database has approximately 53,000 clusters. Each of these clusters is supposed to represent a potentially different gene. Since current thinking is that there are approximately 30,000 genes in the human genome, it is likely that many of these clusters belong together. Of the 53,000 clusters, approximately 32,000 contain at least one mRNA and so represent known genes.

- The Gene Indices (GI) at the Institute for Genetics Research (TIGR) [82] are a resource that is similar in scope to UniGene. As with UniGene, the TIGR GI are arranged according to species. The TIGR GI covers more species than UniGene, with 31 animal species, 30 plant species, 15 protist species and 9 fungal species. Also, the TIGR GI includes a greater number of sequences for most of the species that are also represented in UniGene. The TIGR human gene index contains a similar number of sequences to the UniGene human database. However, it is arranged into approximately 180,000 clusters – significantly more than UniGene. As with UniGene, this is much greater than the number of predicted genes in the human genome, so it is likely that this database will change over the next years. Unlike UniGene, TIGR contains consensus sequences for each of the clusters [83]. From the perspective of designing microarrays, this has both advantages and disadvantages. On the positive side, a consensus is a higher quality sequence and is therefore a better starting point for oligonucleotide design. On the negative side, the UniGene sequences are all real clones and can be purchased from the IMAGE Consortium [84] for use with a spotted array. TIGR also intends to include full information about splice variants in their database. In February 2005, there was very limited splice variant information in the TIGR GI, and no information on human splice variants. This will probably change in the next couple of years and, if implemented, will make the GI a powerful resource for microarray design.

- The third secondary database resource we describe for the construction of microarray is the NCBI's reference sequence project, or RefSeq [85]. The reference sequence project aims to collect high-quality, well-annotated sequenced of many types, including complete genomes, complete chromosomes, genomic regions, mRNAs, other types of RNA, genome contigs and proteins. The

mRNA chapter of RefSeq is of particular interest for microarray design and is available for humans, mouse, fruit fly, rat and zebrafish. RefSeq does not provide a complete picture of expressed material for any of these species [86]. For example as of February 2005, there are only about 19,000 RefSeq entries for humans compared with about 32,000 UniGene clusters containing at least one mRNA. However, the sequences in RefSeq represent the highest possible quality mRNA sequences in the database, and so they are used where possible as the basis for microarray and other work. Splice variants of genes are fully represented in RefSeq, making it a very powerful resource for the design of arrays for known splice variants of known genes.

2.2 Genomic Databases Resources

Genomic databases allow us to examine sequences for microarrays from a genomic perspective: to start with the whole genome and then choose gene sequences for the array based on the annotation of that genome. For small organisms, such as bacteria and yeast, this is the most natural approach. But even for complex organisms such as humans, there exist resources that allow this approach to microarray design and annotation. The main genomic resource for complete genome experiments is Ensembl. Furthermore, some specialized databases exist, for complete microbial genome studies.

Ensembl [87] is a joint project between the European Bioinformatics Institute (EBI) and the Wellcome Trust Sanger Institute to provide complete annotation of eukaryotic genomes. Originally established to cover the human genome, at the time of this writing it also included coverage of mouse, rat, zebrafish, fugu and mosquito. The reason for setting up Ensembl is to provide a single, seamless resource for querying and mining completed genomes, such as the human genome. When a genome is sequenced, it is sequenced in small chunks. Ensembl assembles these chunks into chromosome sequences so that each chromosome appears as a single virtual sequence, also known as the “Golden Path”. The real power of Ensembl as a resource for microarray design is in its annotation [88]. The Ensembl project links all available data about human sequences, so that information on known genes, known proteins and ESTs are included as part of the genome annotation. It also provides annotation on the results of gene prediction algorithms. This is important for microarray design because it allows oligonucleotide probes to be designed for predicted genes and exons in addition to known expressed sequences.

Microbial genomes are small – typically with genomes between 2 and 5 megabases, and between 2,000 and 5,000 genes. This makes microbes very attractive organisms for microarray analysis: it is possible to place probes for every gene in the organism on a single array and perform powerful experiments.

Microbial genomes are readily accessible from two databases: GenBank and the TIGR Comprehensive Microbial Resource (CMR) [89]. In December 2002, there were 102 genomes in GenBank and 96 genomes in TIGR. Data are exchanged between the two databases: most genomes are in both database, but the genomes that are sequenced in TIGR are published in the TIGR database before they reach GenBank, and genomes sequenced elsewhere are published in GenBank before they reach TIGR. Of the 102 genomes in GenBank, there are 85 different organisms, with 12 organisms having multiple strains in the database. The two databases have different

annotation for the same genomes. As a result, an array built from the sequences downloaded from each of these data resources may have slightly different genes.

2.3 The Rise of Post-Genomics: Microarrays and Gene Expression

Gene Expression is defined as the use of quantitative RNA (mRNA)-level measurements of gene expression in order to characterize biological processes and elucidate the mechanisms of gene transcription. The objective of gene expression is the quantitative measurement of mRNA expression particularly under the influence of drug or disease perturbations [90]. As described in [91] the identification of differential gene expression associated with biological processes is a central research problem. High throughput gene expression assays enable the simultaneous monitoring of thousands of genes in parallel and generate vast amounts of gene expression data. The large-scale investigation of gene expression attaches functional activity to structural genetic maps and therefore is an essential milestone in the paradigm shift from static structural genomics to dynamic functional genomics.

2.3.1 Gene Expression Databases: Representing and Sharing Microarray Data

Gene Expression databases provide integrated data management and analysis for the transcriptional expression data generated by large-scale gene expression experiments. Conceptually, a gene expression database can be regarded as consisting of three parts: the gene expression data matrix, gene annotation and sample annotation (figure 6). Samples interfering in a microarray experiments are commonly called biomaterials. In many respects gene expression databases are inherently more complex than sequence databases.

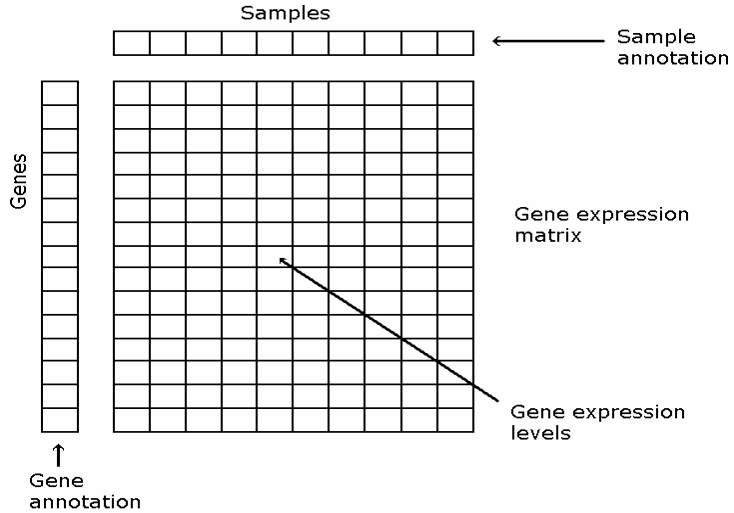


Figure 6. *An abstract form of the annotations of a gene expression matrix.*

As we have mentioned microarray experiments generate large amounts of complex data. Our main purpose is to integrate and share these data among our close laboratory space and the scientific community in general. Except from the obvious reasons about the benefits of the scientific community, by sharing our data there are some additional advantages that we expect to gain.

Firstly we can verify the results of a published microarray experiment hence it is necessary to provide sufficient information so that others can reproduce it. Except

from verifying we may expect from external researchers to perform further experimental work based on the results, expanding and improving our findings. Moreover scientists can compare the results with other functional genomic data. It is valuable to make comparisons either between different microarray experiments or between microarray data and data from other sources (e.g. proteomics). Finally by sharing our data we allow bioinformatics researchers to develop novel data analysis methods.

Data sharing for microarray experiments involves many complications that need to be settled. A microarray laboratory will typically run a number of different computer applications to capture, store, publish and analyze microarray data (figure 7). In order for the laboratory to operate successfully, each of these computer applications, further on referred as components, needs to be able to exchange data with the other. Data should flow seamlessly between the different components, and ideally it should be possible to replace any component without affecting the other parts of the flow. In brief, these components are:

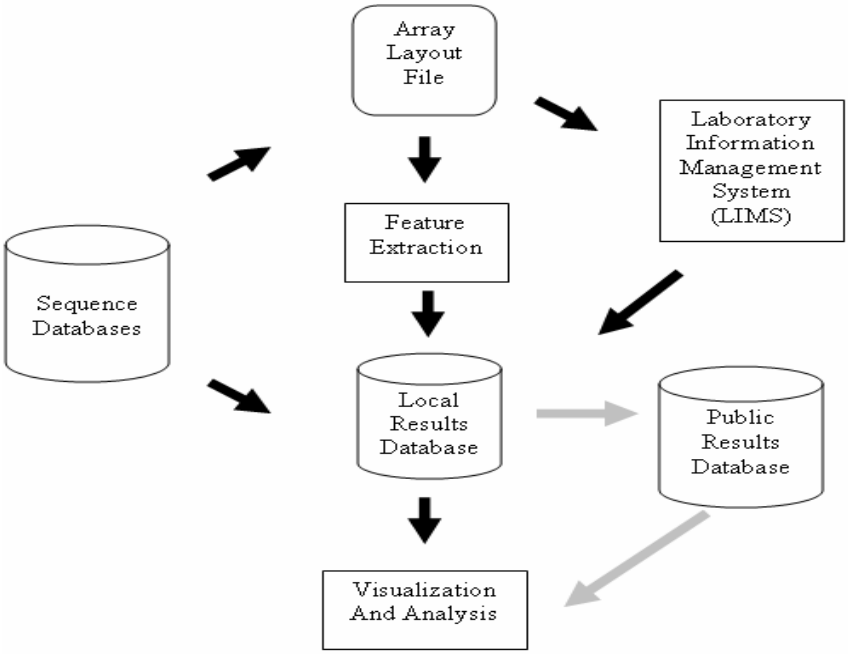


Figure 7. *Software components for microarray data representation and handling.*

- *Array Layout* File is a file containing details of what sequences and genes each feature represent. There are currently many formats for these files, depending on the platform used.
- *Sequence Databases* contain information about the genes that the microarray is measuring and the sequences from which the sequences on the array derive. Accession numbers are included in the array layout so that is possible to connect to these databases.
- *Feature Extraction* Software converts the image of the microarray from the scanner into quantitative information about gene expression. It needs information from the array layout file to be able to annotate the features.

- *Laboratory Information Management System* (LIMS) records all information about the laboratory methods and protocols used in microarray manufacture, sample preparation and labeling, hybridization and washing.
- *Local Results Database* contains results of experiments performed at the local institution. It can be in the form of a formal database or data warehouse, or the data might be stored as information stored in the LIMS.
- *Public Result* Database contains results of microarray experiments that have been published in the public domain. If appropriate, data from the local results database might be transferred to a public database.
- *Visualization* and *Analysis* software allows the user to look at and interpret microarray data. The data could be from the local results database, public data, or a combination and comparison of the two.

Laboratory Information Management System (LIMS). A LIMS records all information about the laboratory experiment, including all procedures, protocols and methods in microarray manufacture, sample preparation, labeling and hybridization. It can be thought of as a laboratory notebook, but with two major benefits. The first is that LIMS can be used to record every step of the experimental process as it happens, including identity of experiment, date, protocols used, and any experimental parameters. The advantage of tracking data are that it provides quality control, as any problems can be traced back to the source. It also provides data reproducibility. If the entire experimental process has been recorded, it is possible for other scientists to reproduce the experiment. Moreover data comparison is provided. By knowing all parameters of the experiment, it is more meaningful to make comparisons between different microarray experiments and to know when comparisons are less meaningful. Finally, data, can be easily published.

If the LIMS system is MIAME-compliant (a universally accepted standard for modelling microarray experiments and representing gene expression data; presented in the sequel), then it will record all the information necessary for publishing the data in a MIAME-compliant microarray database. Another benefit of LIMS is that is possible to include standard protocols as workflows that can help ensure that all staff in the laboratory or group of cooperating laboratories follow the same protocol. This helps to standardize microarray experiments performed by several people.

2.4 Modeling Microarray Experiments: The Standards

Standards are essential for designing computer software that can integrate with other applications by common data representation and information exchange. Especially for microarray databases, in order for standards to be successful, they need to have several qualities. First they have to be useful, flexible and comprehensible in order to accommodate all types if microarray experiments and data, including experiments that have not yet been thought of. They have to be consensual in the sense that they should be agreed upon by microarray users. Finally they have to be straightforward for programmers to implement. In order to be of global benefit, the standards should be adopted by as many research groups and commercial organizations as possible. To achieve this, it is expected that a requirement for the publication of microarray results will be the submission of data to a public domain database that has adopted the standards.

Gene expression data have meaning only in the context of the particular biological sample and the exact conditions under which the sample were taken. For instance, if we are interested in finding out how different cell types react to treatments with various chemical compounds, we must record unambiguous information about the cell type and compounds used in the experiment. The microarray technology is still rapidly developing, therefore it is natural that currently there are no established standards for microarray experiments and how the raw data should be processed. There are also no standard measurement units for gene expression levels. In the lack of such standards the information about how exactly the gene expression data matrix was obtained should be kept in the database, if the data are to be properly interpreted later consequently this complicates the data object model. A common data exchange format MAGE-ML [92] is being developed in collaboration between MGED and some major microarray companies. Most known repositories for gene expression data are ArrayExpress, GEO and BASE.

Microarray data standards comprise three areas. The first is which aspects of the microarray experimental process and of the microarray data need to be recorded. This is the aim of MIAME (Minimal Information about a Microarray Experiment). The second is how to describe the experimental methods and microarray data. For this, we need ontologies defined for our specific domain as controlled vocabularies and relationships to describe genes, samples and data. The third is how to implement MIAME and ontologies in computer software. This requires object models, exchange languages and language-specific modules.

The Microarray Gene Expression Data Society (MGED). The need for microarray data standards was recognized relatively early in the microarray community. In November 1999, the Microarray Gene Expression Data Society (MGED) [93] was founded from EBI researchers, with the intention of establishing standards for microarray data annotation and to enable the creation of public databases for microarray data.

The MGED board of directors and advisory board now has representation from many of the major institutions involved with microarrays, including research institutes, universities, commercial organizations and journals.

MGED has an annual meeting at which major developments are discussed and arranges regular workshops, tutorials and programming jamborees. MGED's work is arranged into four working groups:

i. *MIAME*. Minimal Information About a Microarray Experiment formulates the information required to record about a microarray experiment in order to be able to describe and share the experiment.

ii. *Ontologies*. Determine ontologies for describing microarray experiments and the samples used with microarrays [94].

iii. *MAGE*. Formulates the object model (MAGE-OM), exchange language (MAGE-ML) and software modules (MAGE-stk) for implementing microarray software.

iv. *Transformations*. Determines recommendations of describing methods for transformations, normalizations and standardizations of microarray data.

2.4.1 Minimal Information About Microarray Experiments (MIAME)

The aim of MIAME [95] is to outline the minimum information that should be recorded about a microarray experiment so that data can be fully understood and the experiment fully reproduced in another laboratory. MIAME is intended to assist the exchange of microarray information between researchers, including doing so via the development of public microarray data repositories. It is not intended to be a formal specification, but a set of guidelines. However, it has become the standard for many microarray software packages and databases, so it is highly recommended that we should record data from our experiments in a way that is compliant with MIAME.

MIAME is arranged into two broad areas: the array design description and the experiment description. The reason for this is that the array design is frequently independent of the experiment, with the same array design being used for many experiments.

The aim of the array design description is to give a detailed description of the array, including physical factors (size and material), chemical factors (type of attachment) and logical factors (sequences). To describe the sequences on an array, MIAME introduced three terms:

- *Feature*. The location on the array containing the DNA sequence, also commonly referred to as spots.
- *Reporter*. The DNA sequence on a feature.
- *Composite sequence*. The gene sequence from which the reporter derives. There could be several different reporter sequences for the same gene.

A detailed description of MIAME guidelines can be found at Appendix A.

2.4.2 MicroArray and Gene Expression (MAGE)

MicroArray and Gene Expression (MAGE) [96] is the technical implementation that allows software to be developed using MIAME. MAGE is of interest to researchers seeking to develop microarray software that is fully supportive of MIAME. It is maintained by a group controlled by the MGED society.

This group tries to build software tools capable to facilitate the exchange of microarray information between different data systems. Currently they are doing this through OMG (Object Management Group) [97] by the establishment of a data exchange model (MAGE-OM MicroArray Gene Expression - Data Model [98]) and a data exchange format (MAGE-ML: MicroArray Gene Expression – Markup Language [99]). MAGE-OM has been modeled using the Unified Modeling Language (UML [100]) and MAGE-ML has been implemented using XML (eXtensible Markup Language [101]). MAGEstk (or MAGE software toolkit) is a collection of packages that act as converters between MAGE-OM and MAGE-ML under various programming platforms.

MAGE-OM specifically attempts to define the objects of gene expression data independent of any implementation. Further, tries to abstract the ideas so that the model might be applicable to a broader set of array style experiments. For example, rather than use hybridization, the general class is BioAssay of which hybridization is

a subclass of BioAssayCreation. MAGE-OM can also be used to map to data structures in different platforms, such as Java, Perl, or C++.

An outline of the MAGE-OM workflow, and its component classes are shown in figures 8 and 9, respectively; for a description of MAGE-OM classes refer to table 1.

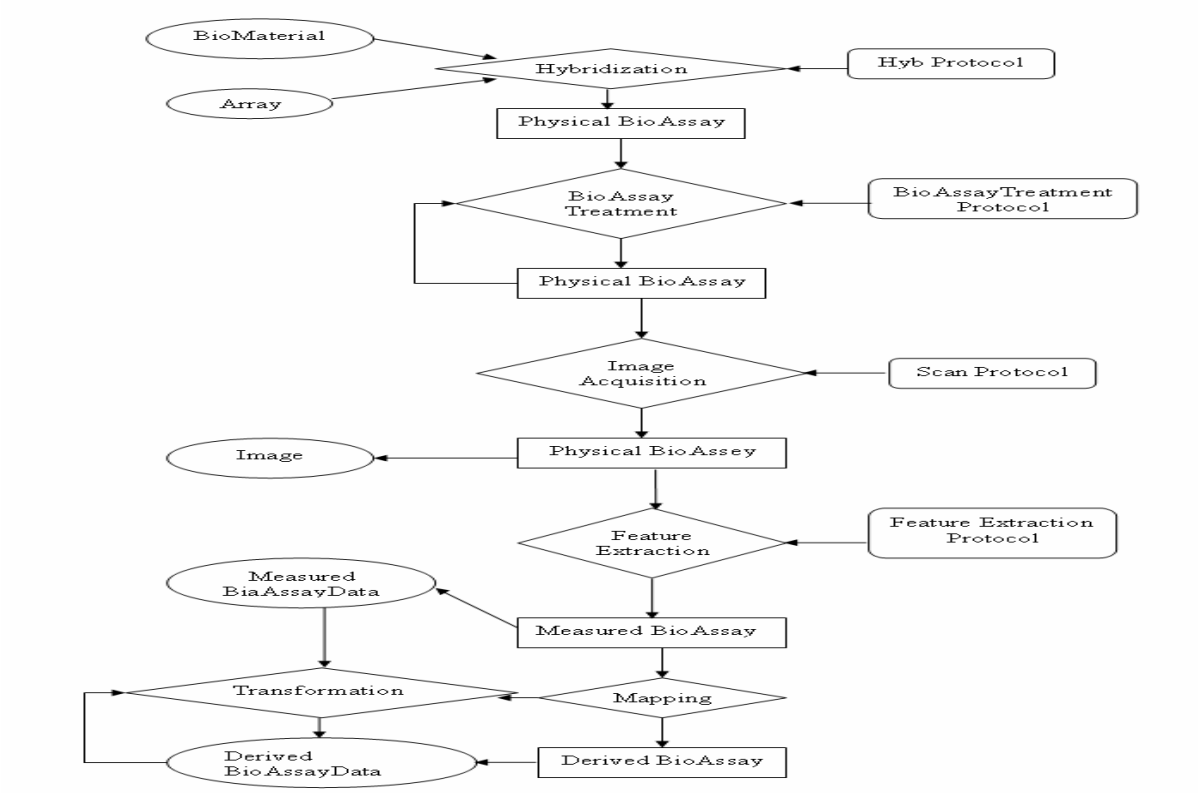


Figure 8. *MAGE-OM Inherent workflow.*

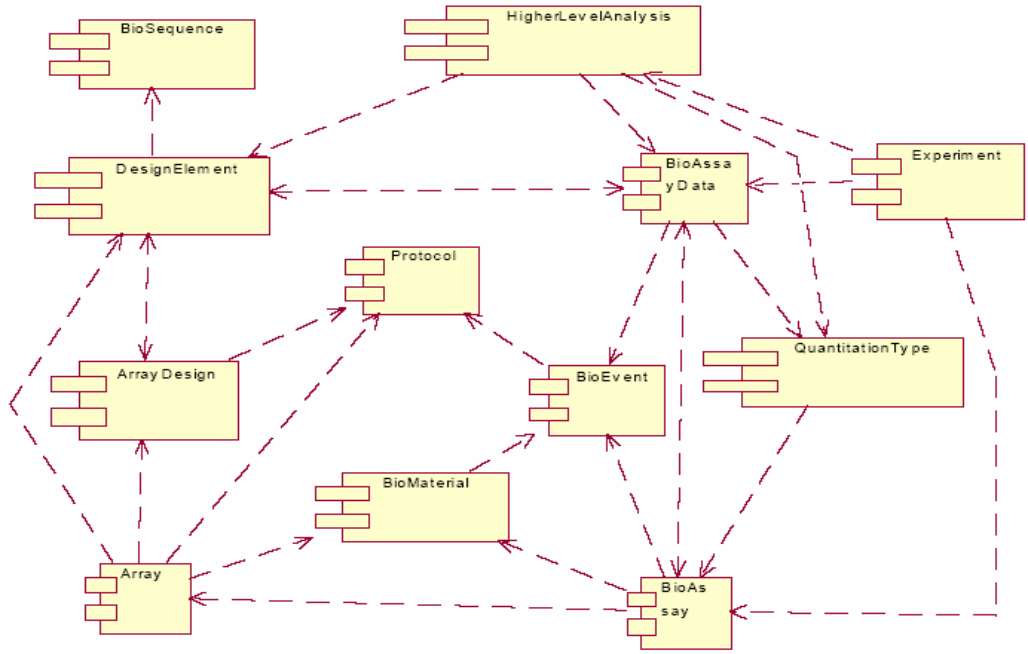


Figure 9. *Main classes of MAGE-OM*

Table 1. *Description of MAGE-OM main classes.*

BioSequence	*Specifies classes that describe the sequence information for a BioSequence.*
QuantitationType	*This package defines the classes for quantitations, such as measured and derived signal, error, and pvalue.*
ArrayDesign	*Describes a microarray design that can be printed and then, in the case of gene expression, hybridized.*
DesignElement	*The classes of this package are the contained and referenced classes of the ArrayDesign and describe through the DesignElements what is intended to be at each location of the Array.*
Array	*Describes the process of creating arrays from array designs.*
BioMaterial	*Specifies classes that describe how a BioSource is treated to obtain the BioMaterial (typically a LabeledExtract) used to create a BioAssay.*
BioAssay	*Specifies classes that contain information and annotation on the event of joining an Array with a BioMaterial preparation, the acquisition of images and the extraction of data on a per feature basis from those images.*
BioAssayData	*Specifies classes that describe the data and information and annotation on the derivation of that data*
Experiment	*Represents the container for a hierarchical grouping of BioAssays.*
HigherLevelAnalysis	*Describes the results of performing analysis on the result of the BioAssayData from an Experiment.*
Protocol	*Provides a relatively immutable class, Protocol, that can describe a generic laboratory procedure or analysis algorithm, for example, and an instance class, ProtocolApplication, which can describe the actual application of a protocol.*
Description	*The classes in this package allow a variety of references to third party annotation and direct annotation by the experimenter.*
AuditAndSecurit	*Specifies classes that allow tracking of changes and information on user permissions.*
Measurement	*The classes of this package provide utility information on the quantities of other classes to each other.*
BioEvent	*An abstract class representing an event that takes sources of some type to produce a target(s) of some type.*

Given the massive amount of data associated with a single set of experiments, XML is the best way to describe the data. The use of a Document Type Definition (DTD) allows a well-defined tag set, a vocabulary, to describe the domain of gene expression experiments. It also has the virtue of compressing very well so that files in an XML format compress to ten percent of their original size. XML is now widely accepted as a data exchange format across multiple platforms. Organizations that request these XML streams can use freely available implementations of either of the W3C [102] recommended DOM [103] or the XML-DEV SAX [104] parsing interfaces to create import and export applications. These import and export applications can be tailored for the specific needs of the organization without the need to burden the vocabulary of the XML with specifics of any organization's schema requirements. The DTD is generated from the MAGE-OM with the addition of the transformed representation of the gene expression data in the DTD. Moreover, it is possible to specify queries both in terms of the object model (OQL) and the XML (XQuery [105], XPath [106]).

The MAGE Software Toolkit (MAGE-stk) is a collection of open source packages that implement the MAGE Object Model in various programming languages. The toolkit is meant for users that develop their own applications, and need to integrate functionality for managing an instance of a MAGE-OM. The toolkit facilitates easy reading and writing of MAGE-ML to and from the MAGE-OM, and all MAGE-objects

have methods to maintain and update the MAGE-OM at all levels. What MAGE-stk doesn't implement, is the interface between an application, and the standard way of representing microarray in MAGE-OM (MAGE-ML when in a file) (figure 10). MAGE-stk is available in Perl, Java, C# and Python programming languages.

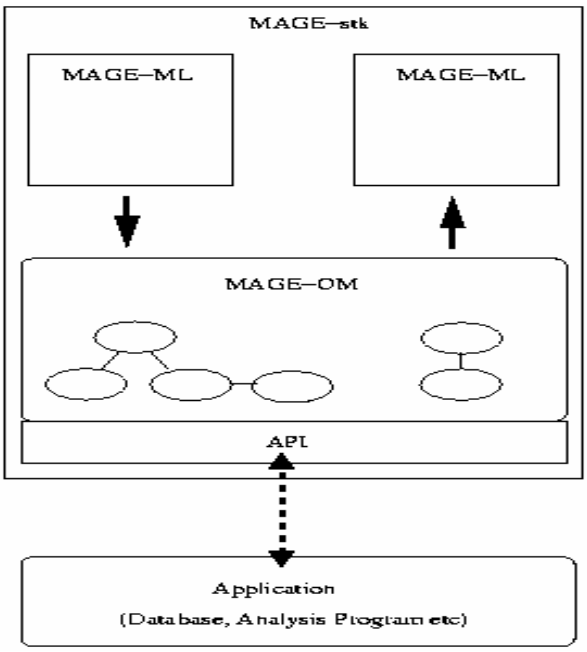


Figure 10. MAGE-*stk offers a API for managing MAGE-OM objects.*

MAGE-OM will store essentially any array based data with very few if any modifications. Applications other than gene expression for which MAGE-OM would be applicable include: protein diagnostics, genotyping, and sectioned tissue analysis. MAGE-OM is not presently capable of storing non-array based expression profiling technologies such as Serial Analysis of Gene Expression (SAGE [107], [108]), and extensive modifications would be necessary to support this.

- ***Other Object Models.*** MAGE-OM stems from the collaboration of some of the major microarray research institutions that contributed by submitting their own object models deployed mostly for inner use. EBI has been working on developing a public repository for gene expression data (ArrayExpress) since 1999 [109]. ArrayExpress development has been centered on ArrayExpress Object Model (AEOM), and design of MAGE-OM has been influenced by this experience. AEOM was mapped to relational tables and implemented as an Oracle 8i database. The National Center for Genome Resources (NCGR) [110] has been developing an open source gene expression database resource, GeneX, since 1999. The GeneX project has focused on the development of a relational data model and a corresponding XML data-transmission model, GeneXML. Finally Rosetta Inpharmatics [111] and Agilent Technologies [112] have been using the GEML 1.0 format as part of internal pipelines. The GEML 1.0 was the object model used to publish the results of sequencing chromosome 22 [113].

2.5 Ontologies

MIAME, details what information is needed to be recorded from a microarray experiment in order to be able to reproduce the experiment. Ontologies provide a solution for how that information can be recorded. The aim of ontology is to give the framework for a formal representation of a subject. An ontology consists of two parts: the vocabulary that contains the words and names of the items in the subject area that are to be described and the relationships that formulates the ways in which the items in the subject area relate to one another.

The main reason for using ontologies is that they help the development of computer databases that hold information about a subject. By introducing a controlled vocabulary, it is possible to query databases using the controlled terms removing any potential ambiguity. Moreover, ontologies provide a conceptual framework that can help in understanding and integrating the information about a subject.

In the field of microarrays there are three sets of ontologies that are used: taxonomic ontologies, gene ontologies and MGED ontologies.

2.5.1 Taxonomic Ontologies

In taxonomic ontologies every living organism is placed in a hierarchy of kingdom, phylum, class, order, family, genus and species. The genus and species together form the scientific name of the organism (e.g., Homo Sapiens). There are controlled vocabularies for each of the terms, and the terms relate hierarchically. Taxonomic databases are rather controversial since the soundness of the taxonomic classifications done by taxonomists so far is directly questioned by the advances of current genomic research.

Various efforts are going on to create a taxonomy resource. Some of them are "The Tree of Life" project [114], "Species 2000" [115], “International Organization for Plant Information” [116], “Integrated Taxonomic Information System” [117]. The most generally useful taxonomic database is that maintained by the NCBI [118]. This hierarchical taxonomy is used by the Nucleotide Sequence Databases [73], Swiss-Prot [119] and TrEMBL, [120] and is curated by an informal group of experts.

2.5.2 Gene Ontologies and the GO Consortium

Gene annotation can be taken care to some extent of by links to sequence databases. Unfortunately, complicated too many relationships between genes in the gene expression matrix and the features (spots) on the array makes it necessary to provide a full and detailed description of each feature on the array, as one gene can relate to several features on the array. The lack of standards in gene naming is another difficulty. A table relating each array feature present in the database to the list of all synonymous names of the respective gene is an essential of a gene expression database.

Gene ontologies provide a set of terms for describing genes and their products. The Gene Ontology (GO) Consortium [121] was set up in 1999 in order to provide a common framework for its members to be able to describe genes and gene products. The consortium members contain major institutions that have serious involvements in gene research for certain organisms. GO has allowed its members to have a common set of terms for annotating genomes. The main advantages to using GO is

that simplifies database querying, makes easier cross-species comparisons and it eliminates any ambiguity in gene descriptions.

GO has organized ontologies for describing genes on three levels. The first is the molecular function level where the task performed by individual gene products is described. The second is the biological process level where the broad biological goal of the gene products (e.g., mitosis or protein degredation) is described. Third is the cellular component level where the subcellular organelle, location or macromolecular complex in which the gene product would operate (e.g., nucleus) is described. Each of these areas has a separate ontology defined for it, and any gene would have terms from all three ontologies.

GO terms and ontology terms in general, exist in a hierarchy of more general and more specific classes. In classical ontologies, each term may only have one parent. However, due to the complexity of biological information, in GO each term can have more that one parent. More precisely, the terms are organized in directed acyclic graphs.

2.5.3 Microarray Ontologies

The Ontologies Working Group at MGED has drawn up ontologies for microarray annotation with the aim of describing microarray data. The MGED ontology comprises three broad type of information: Classes, Properties and Individuals.

Classes are the categories of information, for example age and protocol. Each class has a number of fields describing it. These are:

- *Namespace*. A URL for the ontology
- *Documentation*. A free text description of the class
- *Type*. In the microarray ontologies, every class is of primitive type. This means that the class is not fully defined by its constraints.
- *Superclasses*. The parent classes of which this class is a special case.
- *Constraint*. These are rules by which any single instantiation of the class contains information. Each constraint is in the form of a property that the class may have.
- *Known subclasses*. These are child classes of the class which represent specializations of the class.
- *Used in class*. These are the classes that use this class as part of a constraint.

There is annotation for each for the fields. The superclass MGEDOntology is the root class from which all classes are derived. As protocols are widely used in microarray experiments, there are several constraints that can be used to describe the protocol and many subclasses or classes that contain protocols as a constraint.

Properties encapsulate information about classes. A class has properties. For example the class protocol has the property has_citation. Each property is then linked to a class via the constraint in the class that contains the property. In the case of a protocol, has_citation will take a value in the class BibliographicReference. Properties generally contain less information than classes.

Individuals are instances of classes that are formally included in the ontology. Usually individuals have very little information associated with them.

MAGE-OM contains 226 classes that use 109 properties, and are used to model 644 individuals. It is available as OWL, DAML and RDFS [122].

2.6 Expression Database Comparison

➲ **The first objective of the current thesis was to analyze existing microarray gene expression databases for their ability to serve as an integrated environment for a laboratory that performs microarray experiments. The aim for this comparison was to *choose the most suitable environment* for the experiments performed by molecular biology researchers in the Institute of Molecular Biology and Biotechnology of FORTH (FORTH-IMBB [123]), and informaticians from the Institute of Computer Science of FORTH (FORTH-ICS [124]). The whole effort was to deliver an integrated clinico-genomics environment for the respective experiments in the context of the projects: *PrognoChip* (funded by the General Secretariat for Research & development – EPAN program) and at the same time contribute to the FORTH-ICS efforts in the context of the *INFOBIOMED* (NoE, funded by EU in the context of the IST program).**

In the following sub-chapters we present the two Expression databases that were compared: *ArrayExpress* and *Base*.

2.6.1 ArrayExpress

The EBI (European Bioinformatics Institute [125]) has established a public repository for microarray gene expression data called ArrayExpress [109]], analogous to EMBL-bank for DNA sequence data. ArrayExpress uses MIAME set of disciplines to describe all the information stored. As of November 2004, ArrayExpress contains ~12,000 hybridizations covering 35 species. The majority of studies concern samples from Homo Sapiens or Mus musculus. Along with Gene Expression Omnibus [126] and CIBEX [127], it is one of the three repositories recommended by the MGED society for storing data related to publications. The ArrayExpress suite of databases and applications comprises:

- *MIAMExpress* [128], a web-based MIAME supportive data-submission tool
- *ArrayExpress repository* that provides public and password-protected access to the submitted data
- A *query* optimized data warehouse containing a curated subset of normalized data
- *Expression Profiler* [129], an integrated online visualization and analysis tool.

All the software in the ArrayExpress suite is open source. There are two major submission routes to ArrayExpress: online via the MIAMExpress data submission tool and via a MAGE-ML based pipeline set-up with an external application or database. MIAMExpress is primarily aimed at users with no substantial local bioinformatics support and with no access to a local database providing direct deposition. MIAMExpress is an open source software that can be customized for use by a single laboratory, or for particular application domains.

The highest level of organization in the ArrayExpress repository is the Experiment, which consists of one or more hybridizations, usually linked to a publication. The

ArrayExpress query interface provides the ability to query for Experiments, Protocols and Array designs by their various attributes, such as species, authors or array platforms. The data can also be analysed and visualized online using Expression Profiler. Password-protected access to pre-publication data is provided for submitters and reviewers. A schematic diagram of the software architecture is shown in figure 11, below [130].

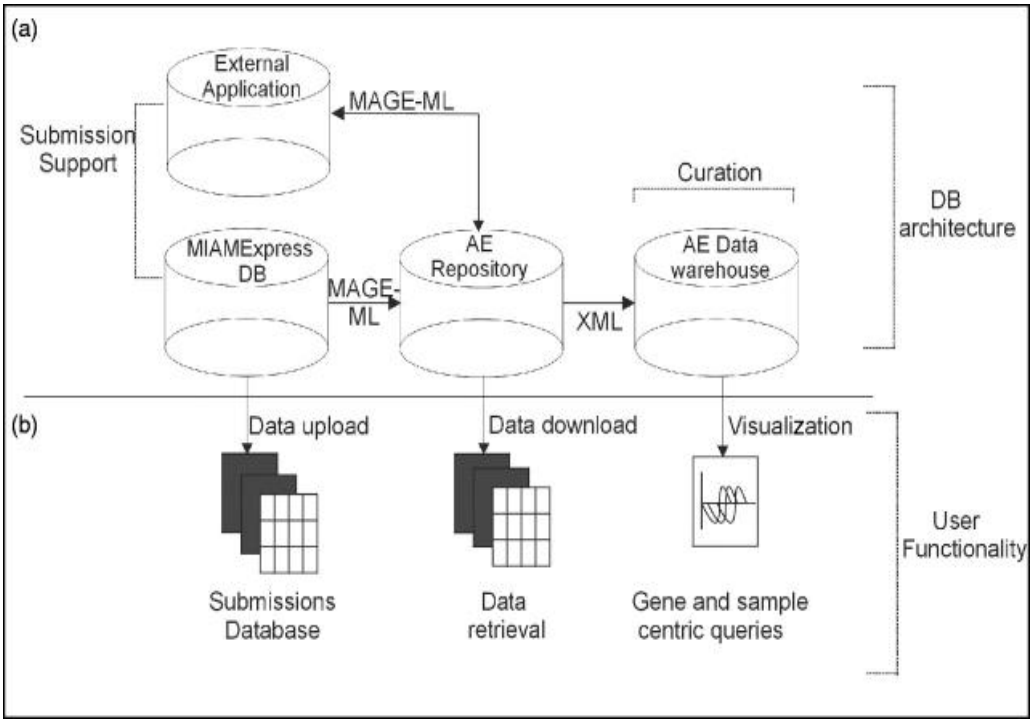


Figure 11. (a) *The ArrayExpress architecture and database side activities are shown. (b). The functionality experienced by the user is shown.*

The online submission tool MIAMExpress is being extended to allow a spreadsheet based data batch uploading to facilitate large-scale experiment submissions. A graph-based visualization tool is being added to MIAMExpress and ArrayExpress. The ArrayExpress repository and data warehouse interfaces will be unified. The gene-based query facility in the warehouse will be used as the basis for integrating ArrayExpress into all EBI services more closely.

2.6.2 BASE: BioArray Software Environment

BASE [131], [132] is a comprehensive free web-based database solution for the massive amounts of data generated by microarray analysis released under the GNU Public License [133]. BASE attempts to be a unified system capable of organizing all the information surrounding microarray experimentation and which also integrate this information with tools for the analysis of quantifies microarray hybridization data. BASE is a MIAME-supportive customizable database and analysis platform designed to be installed in any microarray laboratory and to serve many users simultaneously via the web. The software was developed on the GNU/Linux operating system in the

PHP language [134], with data being stored in a relational database (MySQL [135]) and communicated to the user through the Apache web server [136]. Where needed, the user interface employs Java and JavaScript in addition to plain HTML, and C++ has been used for the more computationally intensive tasks on the server.

The system integrates biomaterial information, raw images and extracted data, and provides a plug-in architecture for data transformation, data viewing and analysis modules. Additionally, for laboratories the system has array production LIMS features that can be integrated with data analysis. The structure of BASE was designed to follow the natural workflow of the microarray biologist (figure 12), and it is compatible with most types of array experiments and data formats. With his or her own account and administrated access levels, a user can import data into the database, group array data together into experiments, and in a uniform and streamlined fashion, apply filters and transformations and run analyses. To facilitate online collaboration users can share almost any object within database. Data can be exported in a multitude of formats for local analysis and publication. BASE also contains an annotating and tracking system for biomaterials that is user customizable via a web interface and is integrated with the data analysis. Source organism and cell-type taxonomies can be created, and new annotation types can be defined and linked to any sample.

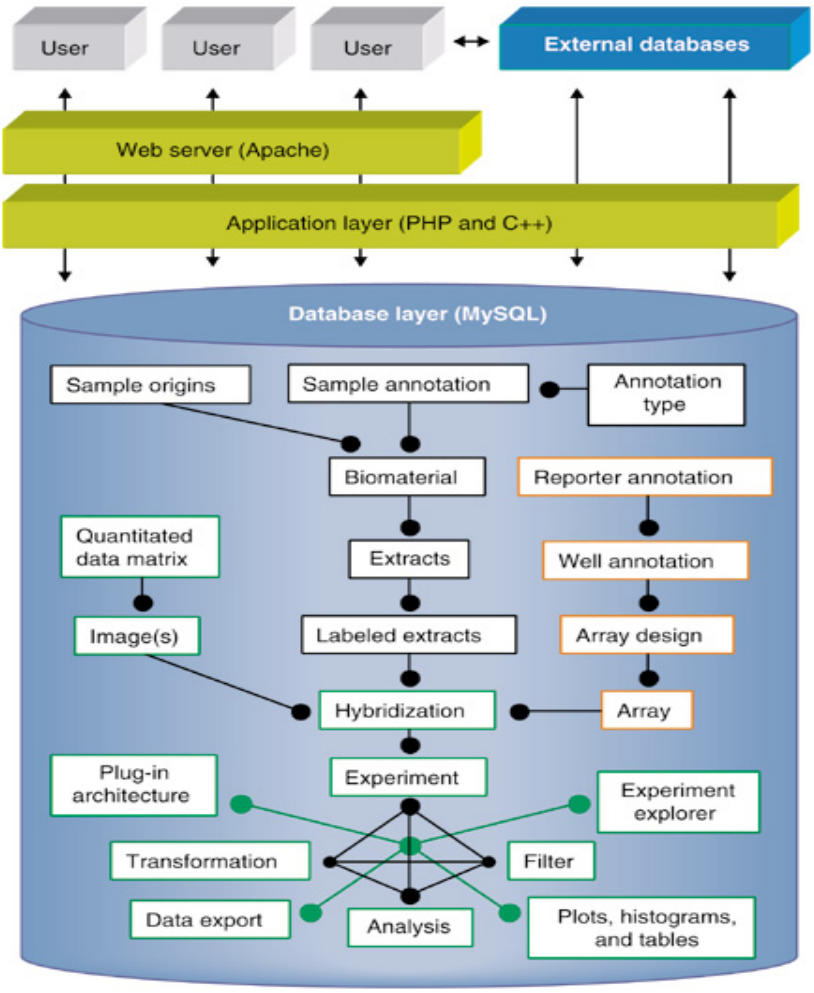


Figure 12. *Simplified schematic overview of software structure of BASE.*

BASE integrates a framework with a plug-in architecture that enables integration of modules that transform, or analyze and visualize microarray data. This architecture consists of three parts: a data standard and format (MAGE-ML) for transferring biomaterial, reporter and hybridization data to and from application modules that run on the server, a job handler for execution of application modules and saving results back into the database, and a web interface for the administration and installation of new plug-in modules. Furthermore, to allow for any combination and series of data filtering, transformation and number-crunching steps, a data analysis interface that is organized hierarchically was created. Finally data can be visualized at several stages of analysis. Unmodified and transformed data sets can be plotted as scatter plots, histograms or tables. Data can also be exported for custom analyses and local development of new analysis methods, and in various defined formats for use in external analysis programs.

2.7 ArrayExpress vs. BASE: Comparison Outcome

During this comparison we evaluated a variety of aspects from the sequence databases ArrayExpress and BASE. The aspects evaluated and our estimations were focused on the different functionalities supported by the two approaches and systems.

- ***Supporting Standards.*** **Both databases provide partial support for experiments described under MIAME guidelines as well as provided data exchange through MAGE-ML. ArrayExpress seemed to have certain difficulties in using MIAMExpress as experiment submission tool. At the other hand BASE had as future plan to provide MAGE-ML experiments submissions even though it supported experiments MAGE-ML extractions.**
- ***Well-known, supportive community.*** **The databases should be well-known and tested under various conditions. There should be also a community of developers/testers that should provide support and instructions whenever we faced any problem. Both databases had mailing lists, and an active community to help and support. ArrayExpress as the sequence database of one of the three major genomic research institutions worldwide had better support, documentation and on-line help.**
- ***Installation and software maintenance.*** **The databases should be relevant easy to install, upgrade and maintained. There also shouldn't have extreme hardware requirements. At this criterion ArrayExpress had serious disadvantages. As an internal Relational Database Management System (RDBMS), ArrayExpress used Oracle version 9i. Even though this RDBMS is one of the most expert and fast it was exceedingly tricky to be installed and set-up. Moreover it had unnecessarily high hardware requirements. At the other hand BASE was depending in MySQL as internal RDBMS, that is light-weighted robust and easy to be installed database system.**
- ***Provided tools / Extensions.*** **The databases should have inherently a large collection of data analysis tools and there should be easy to be extended with new ones. BASE provided some basic data analysis tools**

and an integrated plug-in schema, while ArrayExpress provided just some basic tools. More over ArrayExpress has been built by using Perl programming language rather than BASE that has been built by PHP. PHP is a relevant contemporary language that has more potentials than Perl.

- ***Interface supplied / Usability / Security.*** **As one of the target groups to use the database was not IT experts, the databases should have an intuitively simple but yet functional user interface. Both databases conveyed a graphic query interface. ArrayExpress lacked a graphic submission tool and both databases had paid substantially attention to provide a usable graphic interface. Furthermore, both databases had adopted the security schema of their inherent databases. ArrayExpress had Oracle's security system which is more sophisticated and flexible than the respective RDBMS of BASE.**

➲ **After weighting the pros and cons of each tested microarray gene expression database we finally choose the BASE database. The reasons for this decision were the specific difficulties of ArrayExpress inherent RDBMS installation and maintenance and the flexibility of BASE's plug-ins and PHP developing language. Although ArrayExpress seemed to be more renowned in the bioinformatics community the contemporary characteristics of BASE and its rising reputation determined the final decision.**

3. Combined Clinico-Genomic Knowledge Discovery

Today, the application of novel technologies from proteomics and functional genomics to the study of diseases (e.g., cancer) is slowly shifting to the analysis of clinically relevant samples such as fresh biopsy specimens and fluids, as the ultimate aim of translational research is to bring basic discoveries closer to the bedside [137]. It becomes evident that in order to fully grasp the mechanisms of a disease we do not only need an understanding of the genetic base of the disease - dealing with large amounts of data and related *functional genomics* approaches but we also need to *integrate* the knowledge normally processed in the clinical setting. In other words the research agenda should be forwarded towards the integration or, *synergy* between *bioinformatics* and *medical informatics* activities. In this setting a new discipline namely, *BioMedical Informatics* (BMI), is rising [138], [139] with the vision being to compact major diseases on an *individualized* diagnostic, prognostic and treatment manner [140], [6].

With the recent advances in *microarray* technology [141], [14], the potential for molecular diagnostic and prognostic tools seem to come in reality. The last years, microarray-chips have been devised and manufactured in order to measure the expression profile of thousands of genes. In this context a number of pioneering studies have been conducted that profile the expression-level of genes for various types of cancers such as leukaemia, breast cancer, colon, lymphoma, central nervous system, and other tumours [142], [143], [144], [145], [146], [147], [148]. The aim is to add molecular characteristics to the classification of diseases so that diagnostic procedures are enhanced and prognostic predictions are improved. These studies demonstrate that gene-expression profiling has great potential in identifying and predicting various targets and prognostic factors of diseases.

By measuring transcription levels of genes in an organism under various conditions, in different tissue samples, we can build up *gene expression profiles*, which characterize the dynamic functioning of each gene in the genome. The microarray data are represented in a *matrix* with *rows* representing genes, *columns* representing samples (e.g. various tissues, developmental stages and treatments), and each cell containing a number characterizing the expression level of the particular gene in the particular sample, i.e, the *gene expression matrix*.

Gene-expression data analysis depends on Gene Expression Data Mining (GEDM) technology, and the involved data analysis is based on two approaches: (a) hypothesis testing - to investigate the induction or perturbation of a biological process that leads to predicted results, and (b) *knowledge discovery* - to detect underlying *hidden-regularities* in biological data. For the latter, one of the major challenges is *gene-selection* [149]. Gene selection methods utilise statistical methods and algorithms to estimate the *correlation strength* of genes with any of the sample *classes* or, *phenotypes* (i.e., tumour-types; disease-recurrence states, etc). Genes with high ranked ordered correlation scores will be proposed as possible indicative markers for the targeted phenotypes. Possible *prognostic* genes for disease outcome, including response to treatment and disease recurrence are then selected to compose the *molecular signature* or, *genotypical* category. The selected genes, after tested for their reliability (e.g., via appropriately conducted clinical trials) present the *gene-markers* that are used for the categorisation of new patient samples into respective disease classes.

3.1 Genomic Medicine and Individualized Healthcare

Genomic medicine comes largely from knowledge emanating from the Human Genome Project [1], and has to be embodied in today's healthcare services and research. Genomic medicine will change healthcare by providing knowledge of individual genetic predispositions via microarray and other technologies. Knowledge of individual genetic predispositions can benefit patients in several ways. Firstly by individualized screening, namely by performing certain clinical tests and observations, secondly by individualized behavior changes and by presymptomatic medical therapies. This individualized treatment will introduce the advance of pharmacogenomics that will allow individualized medication use, based on genetically determined variation in effects and side effects. It will also introduce new medications for specific genotype disease subtypes.

- One of the greatest challenges from genomic medicine is to change healthcare by providing better understanding of non-genetic, environmental factors in health and disease, thus by emphasizing health maintenance rather than disease treatment. Moreover we expect from genomic medicine through genetic engineering to explicit or implicit intervene into human genomic sequence. Summarizing, we expect from genomic medicine to change healthcare by creating a fundamental understanding of the etiology of many diseases, even non-genetic diseases.

3.1.1 Applications of Genomic Medicine in healthcare

Although the healthcare community is already applying gene related therapies to specific diseases there are certain limitations to practices followed. The major limitation is that these practices include conditions wholly caused by an extra or missing complete chromosome or part of a chromosome (e.g., Down syndrome, Turner syndrome). Sometimes the conditions are caused by a mutation in a single gene (e.g., cystic fibrosis). These conditions are of great importance to individuals and families with them but, even when added together, are relatively rare. Most people are not directly affected, thus genetics have played small role in healthcare, and in society in general so far. It is indicative that genetic care could be supplied primarily by medical geneticists and genetic counselors, with occasional involvement of other specialists and primary care providers, rather by an organized healthcare system.

In USA, projections suggest that 40% of those alive today will be diagnosed with some form of cancer at some point in their lives. By 2010, that number will have climbed to 50% [150]. Today it is known that 9 of the 10 leading causes of mortality have genetic components making clear that a new perspective of genetics and genomic medicine has to be established [4]. This aspect of genetics has to consider diseases caused partly by mutations in specific genes (e.g., breast cancer, colon cancer, diabetes, Alzheimer disease) or prevented by mutations in genes (e.g., HIV, atherosclerosis, some forms of cancer) [5]. These conditions are also of great importance to individuals and families with them, but are significantly common enough to directly affect virtually everyone. This make genetics play large role in healthcare and in society. Moreover, these conditions are common enough that genetics will be supplied with occasional involvement of medical geneticists and genetic counselors, but primarily by primary care providers and other specialists.

3.2 Biomedical Informatics (BMI): The Vehicle through

Biomedical informatics is an emerging discipline underlying the acquisition, maintenance, retrieval and application of knowledge and information in research, education, and service in health-related basic sciences, clinical disciplines, and health care administration with computer science, statistics, engineering, mathematics, information technologies and management.

Biomedical informatics coalesces the related fields of *Medical Informatics* (now being named Health Informatics) and *Bioinformatics*. Health Informatics contains subsets such as Telemedicine, Clinical Informatics, Pharmaceutical Informatics, Nursing Informatics and Public Health Informatics.

Central to both medical informatics and bioinformatics is the collection and analysis of information. While medical informatics is more concerned with structures and algorithms for the manipulation of the data and how it can be applied in healthcare, bioinformatics is more concerned with the data itself and its biological implications.

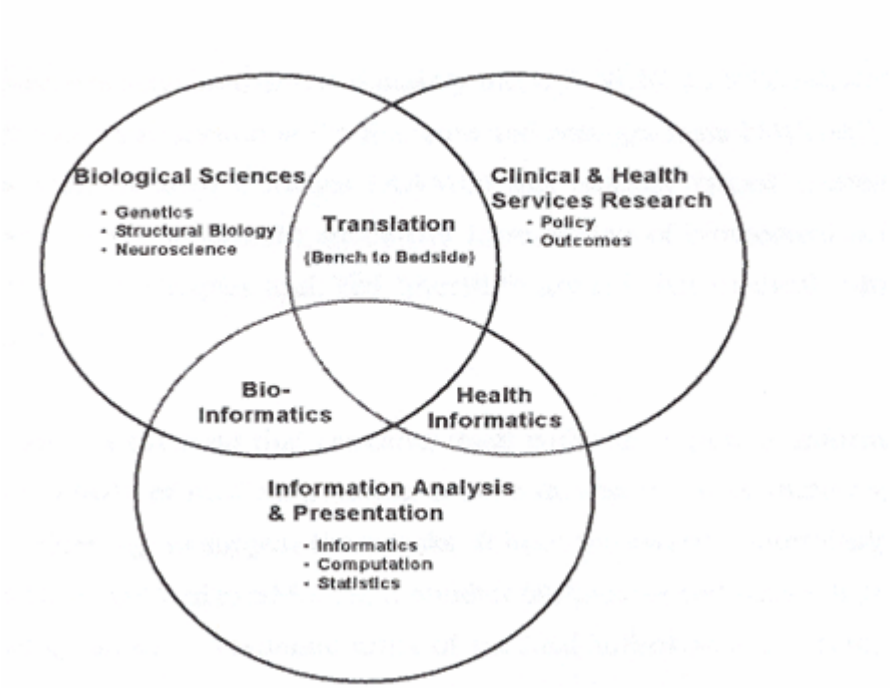


Figure 13. *An Informatics-centric view of biological sciences, healthcare, bio- and medical-informatics, that orients biomedical informatics R&D.*

Figure 13, above, shows an informatics-centric view of the intersections and overlap among the biological sciences, health services research, and information analysis and presentation.

BMI research areas [151] include**:** (1) understanding how and why researchers and practitioners use information to accomplish their objectives; (2) modeling structures for representing data and information that make relationships between concepts and terms explicit; (3) developing and evolving computer-assisted decision support systems to improve clinical practice, biomedical research, education, and administration; (4) understanding and addressing related workflow, change management, communication, and human-computer interface issues; and, (5)

developing methods for evaluation of models and systems, including health services research, data mining and limiting retrieval to context.

3.3 Integrating Clinical with Genomic Information in Gene-Expression data Analysis

Most genetic contributions to common disease identified so far have been low frequency with high penetrance alleles. These alleles include: BRCA1 and BRCA2 (breast and ovarian cancer), HNPCC (colon cancer), MODY 1,2,3 (diabetes), Alpha-synuclein (Parkinson disease). Nevertheless, on a population level, most genetic contributions to common disease are from high frequency, low penetrance alleles. These alleles include: APC I1307K (colon cancer), ApoE (Alzheimer disease), CCR5 (HIV/AIDS resistance) [152]. What makes these low penetrance alleles to be expressed seems to be a complex concept that has to include clinical observations alongside with genomic medicine.

Generally, one major research hypothesis is that clinical observations are strictly correlated with specific alleles during the expression of serious diseases like cancer and diabetes. To identify genetic patterns - in the broadest sense - which are relevant to patients in general, genetic data must be linked with clinical data for a substantial number of patients. While we are moving towards the integration of clinical information along with genomic medicine it is crucial to build information systems, software tools and services that elaborate this integration.

Until recently, diagnostic and prognostic assessment of diseased tissues and tumours relied heavily on indirect indicators that permitted only general classifications into broad histological or morphological subtypes and did not take into account the *alterations* in *individual gene expression.* In this context, global *gene expression analysis* using *microarrays* now offers unprecedented opportunities to obtain *molecular signatures* of the state of activity of diseased cells and patient samples. This groundbreaking approach of studying cancer promises to provide a better understanding of the underlying mechanism for *oncogenesis*, more accurate diagnosis, more comprehensive prognosis, and more effective therapeutic interventions.

Within the past years, two major advances have taken place. First, *microarray-based expression profiling* has shown promise with the preliminary demonstration that clustering techniques can ease re-classification and predict the clinical outcome for various diseases [139], [143], [153], [154]. These studies demonstrate the transition of basic biologic research to clinical application. The predictive power of this approach is much greater than that of currently used approaches, but remains to be validated in prospective clinical studies.

3.3.1 Integrated Clinico-Genomic Knowledge Discovery: A Scenario

The conceptualization of individualized medicine is to be realized by respective procedures, protocols and guidelines in the context of *integrated* and *synergic clinico-genomics decision-making scenarios*. In the following lines an outline of such a scenario is presented for the case of cancer – the same scenario may be conceptualized and appropriately extended to other diseases, The scenario illustrates the key processes, namely: collection of samples, phenotyping, genotyping and the transition from phenotypes to genotypes.

i. *Collections of samples.* Tissue sample is extracted from specific cancer patients. This applies not only to surgical operations (where, the tumor is extracted) but also to cases where the appointed protocol involves a pre-surgical chemo- and/or radio- therapeutic treatment in order to 'shrink' the tumour and then, depending on the outcome, proceed to surgery invasion. The tissue sample is appropriately treated and preserved in order to reserve RNA expression.

ii. *Phenotyping*

   - *Characterization of samples.* Assume that the collected samples are assigned (by the involved clinical specialist – oncologist, pathologo-anatomist, chemo- and/or radiotherapist) to various *clinico-histopathological types* and *stages*.
   - *Classification of samples.* According to characterization, the samples may be assigned to different *phenotypical* profiles (e.g. phenotypes F1 and F2 – see Figure 14). The profiles refer to parameters of patients' clinical assessment and include: age, habits & environmental factors, family-history, tumour type, stage and other related histopathological parameters, as well as medical-imaging parameters. In this case the acquired patients' phenotypes ease the involved diagnostic and/or prognostic decision making operations (e.g., good vs. bad prognosis). In the case of therapeutic decision making, and in the presence of follow-up information, the phenotypes may refer to the potential treatment outcome, e.g., patients (samples) responding to a specific chemotherapeutic and/or radio-therapeutic treatment versus patients that do not respond.

iii. *Genotyping.* Using *microarrays* technology the molecular, i.e. gene-expression, profiles of the samples are extracted. Moreover, based on fundamental molecular biology knowledge we may assess relevant molecular-pathways (e.g., genetic networks). Such knowledge will help to the identification of validated and more refined genotypes.

iv. *From Phenotypes to Genotypes.* After i, ii, and iii are accomplished, we have at our disposal a *gene-expression matrix* with rows the targeted genes and columns the expression levels of genes for the different samples. Moreover, each sample is assigned to one of the two identified phenotypes, F1 and F2, which are *classes*. Applying advanced *data-mining* operations – such as *gene selection*, on the acquired gene-expression matrix we are able to identify potential discriminatory genes, i.e., the genes that distinguish between the two identified phenotypes. These genes compose and indicate the *molecular signature* (or *gene markers)* of the respective phenotypes or, the most discriminatory *features* that best distinguish between the classes. In other words, we are able to link potential phenotypical profiles to respective molecular or, *genotypical* ones. Such advancement may be utilized in the course of both prognostic and therapeutic decision-making processes. That is, respective patients, whose gene-expression profiles 'match' the discovered molecular signature, could be detected to belong to one of the identified phenotypes. Then, according to assessed prognostic indicators and established clinical guidelines the respective patients may be admitted to (potentially) available treatment protocols.

v. *From Genotypes to Phenotypes.* The scenario presented in iv demonstrates the identification of patients' populations that best 'fit' specific molecular profiles and by though, ease the individualized treatment/care objective. The decision making

process described above may be initiated the other way around, towards the establishment of more *fundamental* knowledge. That is, applying again data-mining operations (e.g. clustering) we are able to identify clusters of samples based on their gene-expression profiles. These clusters (actually the ones validated by the involved researcher) may represent potential interesting genotypes, e.g., genotypes G1 and G2 (figure 14). So, in the course of diagnostic, prognostic or, therapeutic decision making process, each, yet untreated, patient may be assigned to its corresponding genotypical class (i.e., to the discovered cluster genotype into which the patient belongs). Then, with the aid of a supervised predictive learning operation (for instance, decision trees) re-classification of the disease on the phenotypical level - a fundamental task in the clinical research for compacting major diseases.

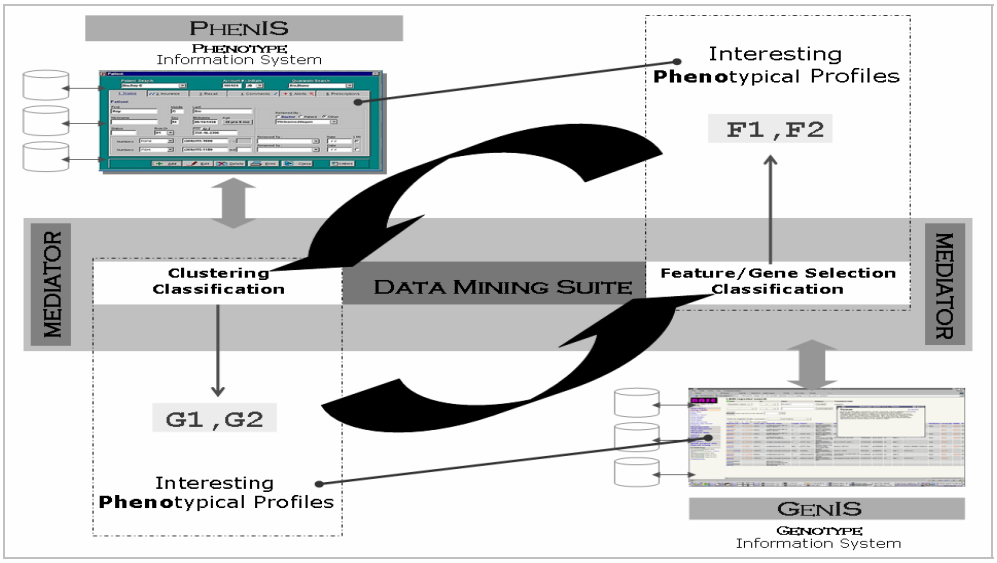


Figure 14. *Integrated clinico-genomic knowledge discovery: From phenotypes to genotypes and vice-versa.*

The operationalism of the aforementioned scenario calls for the integration of both clinical and genomic data. Such an endeavor demands the elaboration and customization of a *mediation* infrastructure as well as data mining operations with the appropriate biomedical informatics support. In the heart of such an integrative environment the gene selection processes plays the most important role (figure 14).

3.4 Enabling Infrastructure: Integrated Clinico-Genomics Environment

With the recent advances in microarray technology, the potential for molecular diagnostic and prognostic tools seems to come in reality. In such an integrated environment, the need to extend the standard clinical decision-making references to reliable genomic establishments also raises as a major demand.

While the focus of BioInformatics- BI around the issues surrounding the Human Genome Project has given a scientific strength to BI research and development, the shift to develop clinical applications could produce the same problems that Medical

Informatics- MI professionals have faced during the past decades. New collaborative efforts between MI and BI could provide new insights and create a synergy for challenges needed to create novel genomic applications in medicine [155]. BI enables us to understand the fundamental knowledge about biological processes.

The inclusion of clinical information in biomedical informatics opens the gateway to genetic risk profiling of patients, new paradigms in disease diagnoses and prognoses and novel approaches to drug discovery based on the correlation of genetic and molecular knowledge of diseases with clinical information of the patients. At the same time, it becomes evident that in order to fully grasp the mechanisms of a disease we do not only need an understanding of the genetic base of the disease- dealing with large amounts of data and related functional genomics approaches (such as gene-expression profiling) but we also need to integrate the knowledge normally processed in the clinical setting.

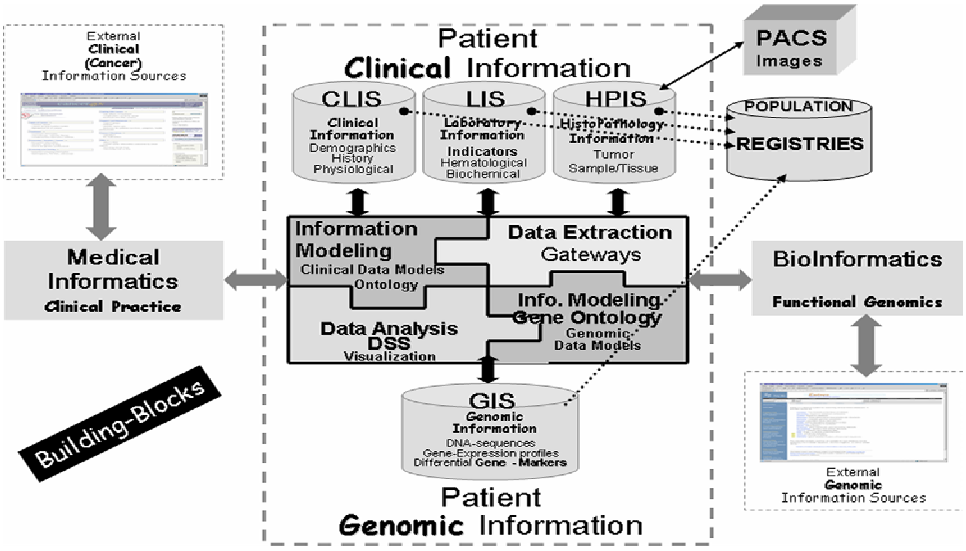


Figure 15. *The envisioned Integrated clinico-genomic environment – knowledge discovery and data mining are key-components of the environment.*

The aim is the design, development and deployment of an integrated clinico-genomics operational framework where, functional genomics and disease compacting research are coupled and guided by related medical knowledge. The endeavour is to be based on the synergy between Medical Informatics and Bioinformatics, and centred on the promising microarray technology. In this setting, the respective R&D agenda should be forwarded towards: the delivery of an *Integrated Clinico-Genomics Environment* – ICGE with the combined genetic- and individualized-medicine being the target. Figure 15, above, shows a general outline of the envisioned ICGE. Key components of the envisioned ICGE environment are: information and data integration (phenotypical and genotypical), and knowledge discovery and data mining operations.

In chapter 2 and 3, the specific contributions of the current thesis to the integration issues were presented. In the following chapter we present specific contributions for related *knowledge-discovery* issues and in particular to *gene-selection* and *clustering* of gene-expression data.

4. Towards Reliable Gene-Markers: Supervised Gene Selection

In this chapter we firstly justify the general concept of supervised gene expression database mining, research pathway and the related work. Then we propose a novel gene selection methodology based on the application of an entropic metric for gene discretisation. The algorithm is composed by four main modules: gene ranking, gene grouping, consecutive feature elimination and class prediction. Furthermore, we apply the algorithm in real-world datasets and we perform a comparison survey based on the resulted accuracy and feature elimination of our method versus other related methods.

4.1 Gene Expression Data Mining

4.1.1 Background to Gene Selection from Microarray Data

Computational genomics has identified a classification of three successive levels for the management and analysis of genetic data in scientific databases: Genomics, Gene expression and Proteomics [156]. In this chapter we will be concerned about Gene expression database mining. Gene expression database mining is the identification of intrinsic patterns and relationships in transcriptional expression data generated by large-scale gene expression experiments.

Gene expression database mining is used to identify intrinsic patterns and relationships in gene expression data. Traditionally molecular biology has followed so-called reductionist approach mostly concentrating on a study of a single or very few genes in any particular research project. With genomes being sequenced, this is now changing into so-called systems approach. Research questions such as how many genes are expressed in different cell types, which genes are expressed in all cell types, what are the functional roles of these genes, how a group of genes is regulated and what genes are interfered in a specific phenotype can now be posed.

Microarray gene expression experiments are organized in four basic types of experimental protocols: a comparison of two biological samples, a comparison of two biological conditions, each represented by a set of replicate samples, a comparison of multiple biological conditions and analysis of covariate information. By ‘biological condition’ we mean the cell or tissue type or variant, plus the environmental or experimental variable that a given sample represents. The environmental or experimental variable may include temperature, exposure to some stimulus, insult, or treatment, or elapsed time from the exposure. These variables may define groups implicitly, or can be defined explicitly as covariates [157].

Although biological experiments vary considerably in their design, the data generated by microarray experiments can be viewed as a matrix of expression levels, organized by samples versus genes. Each sample represents separate microarray hybridization and generates a set of M expression levels, on of each gene. We call this set of expression levels an ‘expression signature’, although the term ‘expression fingerprint’ has also been used. In an analysis we may consider N such samples. For each gene, we can consider its set of expression levels across the different samples, called its expression profile. Outside this matrix of expression levels, we may have covariate information for samples, genes or both. The goal for microarray data analysis is to make inferences among samples, genes and their expression levels and covariates (figure 16).

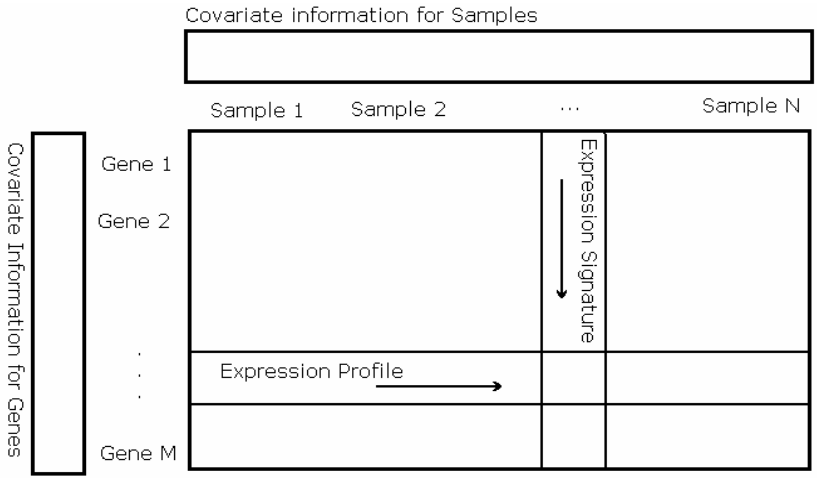


Figure 16. *The Gene Expression data matrix – as resulted from microarray experiments.*

We make a distinction between two types of analysis tasks: *gene selection* and *gene clustering*. Gene selection implies in identifying specific genes that are expressed differentially in one or more biological conditions by identifying unusual patterns of expression. Gene clustering or gene grouping is useful for understanding common expression patterns and it relies on reducing the complexity of the data by clustering genes into groups and identifying potential co-regulated genes.

4.1.2 State-of-the-art Approaches in Gene Selection from Microarray Data

One of the goals of supervised expression data analysis is to construct classifiers, such as linear discriminants, decision trees or support vector machines (SVM), which assign predefined classes to a given expression profile [158]. For instance, if a classifier can be constructed based on gene expression profiles that is able to distinguish between two different, but morphologically closely related tumour issues, such a classifier can be used for diagnostics. Moreover, if such a classifier is based on a set of relatively simple rules, it can help to understand what the mechanisms involved in each tumour are. Typically, such classifiers are trained on a subset of data with a priori given classification and tested on another subset with known classification. After assessing the quality of the prediction they can be applied to estimate the classification of which is unknown.

Brown et al. [159] have applied various supervised learning algorithms to six functional classes of yeast genes using gene expression matrices from 79 samples [160]. Genes from some of the classes, such as ribosomal proteins and histones, are expected to be co-expressed. For these classes it was achieved a good classification accuracy. Some other functional classes, such as protein kinases, are not expected to have distinct gene expression profiles. It was shown that SVM provides one of the best prediction accuracy for the functional classes that are expected to be co-regulated.

Golub et al. [142] applied neighbourhood analysis to construct class predictors for samples, concretely for leykemias. They were looking for genes the expression of which is best correlated with two known classes of leukemias, acute myeloid

leukaemia and acute lymphoblastic leukaemia. They constructed a classifier based on 50 genes (out of 6817) using 38 samples and applied it to a collection of 34 new samples. The classifier correctly predicted 29 of these 34 samples.

Su et al. [161], made a thoroughness study on the expression profiles of 9198 genes probing for discriminant factors for 11 different tumour types. They calculated a Wilcoxon rank-sum score [162] for each group of tumour samples versus samples from all other groups. The 100 genes with the lowest Ps in each class were ranked based on their predictive accuracy for discriminating one class versus all other using a Support Vector Machine (SVM) classifier [19] and ranked based on the Leave One Out Cross Validation (LOOCV [163]) accuracy. They made confident and accurate predictions for 85% of the test samples.

Van't Veer et al. [143] studied the gene expression profile of 78 breast cancer according to their clinical outcome. In brief, 5000 genes significantly regulated were selected from the 25000 genes on the microarray and ranked according to their correlation coefficient, then a sequentially adding of genes method followed to build a predictive mechanism. They finally build a 20 gene predictor capable to predict 65 out of the 78 patients' clinical outcome.

Pomeroy et al. [148] developed a classification system based on microarray gene expression data derived from 99 patient samples with 4 different tumours of the Central Nervous System (CNS). They applied the Self Organizing Maps (SOMs) algorithm and hierarchical clustering to group data and principal component analysis to reduce the dimensionality of the data. Then they ranked data according to a signal-to-noise statistic and the t-statistic metric. Finally they used the k-NN algorithm [164] as a prediction mechanism for test data. They outperformed an 85% prediction score.

Note that when classifying samples, we are confronted with a problem that there are many more attributes (genes) than samples that we are trying to classify. This makes it always possible to find a perfect discriminator to find if we are not careful in restricting the complexity of the permitted classifiers. To avoid this problem we must look for very simple classifiers, compromising between simplicity and classification accuracy.

4.2 A Novel Gene Selection Approach: Methodology and Algorithms

Here we present a novel gene-selection methodology composed by four main modules:

- Discretisation of gene-expression data;
- Gene ranking;
- Grouping of genes;
- Consecutive feature (gene) elimination, or addition.

Discretisation of gene-expression data compose a data pre-processing that takes as input the gene-expression matrix and output a discretised transform of it [165].

The gene-selction methodology is implemented in the context of an integrated gene-expression data analysis system, named MineGene -- a contribution of the current thesis.

An outline of the introduced gene selection via addition/deletion of genes, named is presented in Figure 17, below.

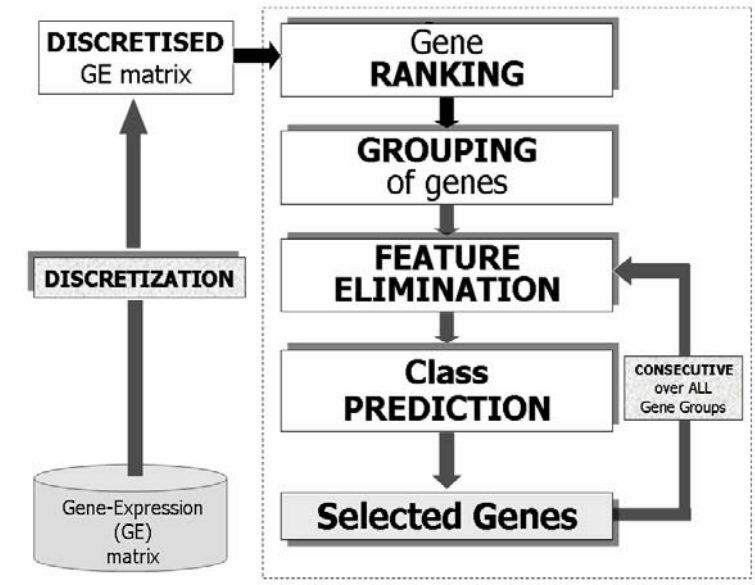


Figure 17. *Outline, components and workflow in the Gene Selection methodology.*

4.2.1 Discretization of Gene-Expression Data

In many gene expression profiling studies the researchers decide to visualize the potential clustering of the genes (or the original gene expression matrix), as well as the final selected set of genes in a discretised manner [142]. Even a simple clustering algorithm based on binning (i.e. discretizing the expression profile space and clustering together the profiles that map into the same bin) has been shown to be useful for clustering genes and subsequent discovering of transcript factor binding sites [166]. The MineGene method utilized discretization of the gene expression continuous values into the core of the gene selection process. Discretisation of gene's expression values leads to the assignment of these values to interval of numbers that bound the expression level of the genes in the given samples. A variable number of such intervals could be utilized and assigned to naturally interpretable values e.g., `low`, `high`.

Given the situation that, in most of the cases, we are confronted with the problem of selecting genes that discriminate between two classes (i.e., diseases, disease-states, treatment outcome, recurrence of disease, in other words phenotypes) it is convenient to follow a *two-interval* discretisation of gene-expression patterns. The multi-class problem where, patient samples are categorised to more than two phenotypes, is tackled by splitting it into a series of two-class discrimination problems and the combining the results, as it is done in various gene-expression studies [161], [167]. In this thesis we also present a novel multi-class categorisation method suited for microarray data.

A general statement of the two-interval discretisation problem followed by a two-step process to solve it follows.

Given: A vector of numbers $V = <n_0, n_2 \ldots n_{v-1}>$, $n_i > n_{i+1}$ where, each number n_i in V is assigned to one of two classes.

Find: A number, μ, $n_i < \mu < n_{v-1}$, that splits the numbers in V into two intervals: $[n_0,\mu)$ and $[\mu,n_{v-1}]$, and *best discriminates* between the two classes – best discrimination is decided according to a specified criterion (in the presented work we rely on an information theoretic one; see step 2 below).

The two aforementioned steps are (see figure 18 for a visual outline of the approach):

Step 1. For all consecutive pair of numbers n_i, n_{i+1} in V their midpoint, $\mu_i = (n_i, n_{i+1})/2$ is computed, and the corresponding ordered vector of midpoint numbers is formed, $M = < \mu_1 , \mu_2 \ldots \mu_v>$.

Step 2. For each $\mu \in M$ the well-known *information gain* metric is computed (utilised in the context of decision tree induction, [210]):

$$IG(V,\mu) = Entropy(V) - \sum_{u \in \{l,h\}} \frac{|V_u|}{|V|} Entropy(V_u) \quad (1)$$

where sets V_l, and V_η include numbers from V which are less than μ and higher (or equal) to μ, respectively. That is, $V_l = \{n_i \in V \,/\, n_i \text{ in } [n_0,\mu)\}$ and $V_h = \{n_i \in V \,/\, n_i \text{ in } [\mu,n_{v-1}]\}$. It is crucial to note that the entropy estimation is made according to the class assignment of each element in V and not according the expression values that it contains. Hence.

$$Entropy(V) = -\frac{|V_{pos}|}{|V|}\log\frac{|V_{pos}|}{|V|} - \frac{|V_{neg}|}{|V|}\log\frac{|V_{neg}|}{|V|}$$

Note that the first term in equation (1) is just the entropy of the original set of numbers in V according to their class assignment, i.e., the distribution of class-values assigned to the numbers in V. The second term is the expected entropy after V is split using μ as the split point. That is, taking into account the distribution of class-values assigned to the numbers in V_l and V_h. The midpoint that exhibits the maximum information gain is considered as the gene's expression value which, when considered as a split point, exhibits the best discrimination between the classes. Then, this point is selected to assign the gene's expression values to the *nominal* '*l*'ow or, '*h*'igh values, respectively (i.e., less than μ and higher that μ). A 'natural' (even extreme and controversial in a molecular setting!) interpretation of low and high expression values for a gene is that the *state* of the gene is '*on*' or '*off*' in a particular sample (e.g., disease type or state).

The aforementioned discretisation process is applied independently on each gene in the training set. The final result is a discretised expression-value representation / transform of each gene. An example, from the leukaemia domain (a two-class/disease discrimination domain between diseases ALL and AML), is shown below (see chapter 4.3).

Gene / Sample-class → ↓	ALL	ALL	ALL	ALL	ALL	ALL	ALL	AML	AML	AML	AML	AML	AML	AML
M77142- original	296	225	243	137	289	-20	150	27	28	45	34	68	80	21
M77142- **discretised**	**h**	**h**	**h**	**h**	**h**	**l**	**h**	**l**	**l**	**L**	**l**	**l**	**l**	**l**

The split values for each gene are stored to be used for (unseen) samples excluded from the training phase. In this case, the expression values of the genes are discretised according to the stored ones. The steps of the overall gene expression discretisation method are presented in Figure 18, below.

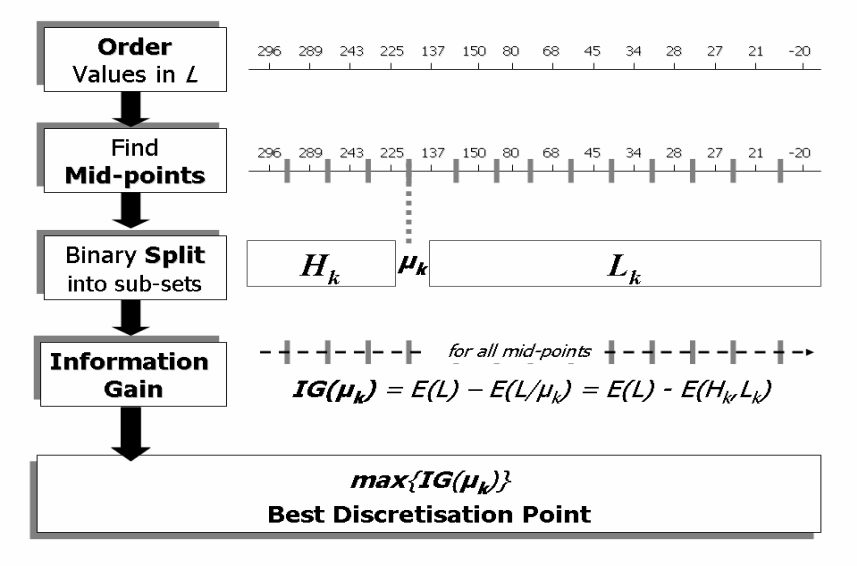


Figure 18. *The gene-expression data discretisation process.*

- *Related approaches.* The discretisation process resembles the one introduced by Fayyad and Irani [168], with two fundamental differences (recently, the same approach was also utilized in a gene-expression profiling study [169]). Because we use the sorted list of numbers for the selection of midpoints, all the points are 'boundary values' (in Fayyad's terminology). Furthermore, in [168] and [169], discretisation is recursively applied to each of the formed binary splits until an appropriately devised stopping criterion is met. The method proposed by Fayyad and Irani, however, does not meet the demand for a two-interval discretisation, which poses a strong difficulty to the natural interpretation of the resulted nominal values as it is unintuitive to interpret the states of a gene that is discretised to more than two values.

4.2.2 Gene Ranking and Selection

The problem that reveals now is how to select the genes that best discriminate between the different classes (being different diseases, disease types/states or, treatment outcome). The problem is well-known in the machine learning community as the problem of feature-selection [170]. In this context various 'wrapper based' [171], or, 'filtering' [172], approaches has been proposed.

Traditionally, in machine learning and data mining research the number of features, m, is quite smaller than the number of cases (samples), *n* that is, $m << n$. In contrast, gene-expression studies refer to a huge number of features and quite few samples. In most of gene-expression domains the number of genes is in the range of 2000 – 35000 (= the estimated number of human genes), and the number of samples in the range of 50 – 200, that is $n << m$. In this context it is questionable if a pure 'wrapper' based feature-selection approach could help, especially because of its high-

computational cost. This argument could be also grounded on the PAC-learnability framework [173]. The needed (i.e., theoretical lower bounds) number of examples for a concept (i.e., a Boolean one) to be PAC-learnable are computed to be *log(m)* [174]-[177]. So, the extra cost of learning a concept in the presence of *r* irrelevant features is *log(m-r)*, the bound, and the computational cost, remains high.

One feature (gene) selection process is based on the combination of a filtering and wrapper consecutive components: (a) *Filtering* component – the genes are *ranked* with respect to their power to distinguish between the different classes, and (b) *Wrapper* component – a *greedy elimination* (or, *addition*) process is consecutively applied on (groups of) the ranked genes in order to select the ones that best discriminate between the classes.

4.2.2.1 Gene Ranking

Gene Ranking has already been used to estimate genes' discriminating ability. Pomeroy et al [148] applied the following metric: each gene, that has samples in class *a* and in class *b* are ranked according to the formula 2, below:

$$\frac{\mu_a - \mu_b}{\sigma_a + \sigma_b} \tag{2}$$

Where μ_a, μ_b are the mean values of the expression values of class *a* and class *b* respectively. And σ_a, σ_b are the standard deviation of expression values of class *a* and class *b* respectively. Intuitively, this formula calculates how 'concentrated' are the expression values among the two classes.

In our approach, for each discretised gene we count the number of '*h*'s and '*l*'s that occur in the respective samples. Assume that each sample is assigned to one of two classes, i.e., *P*, and *N*. The following quantities are computed: $H_{g,P}$ = number of '*h*' values for gene *g* assigned to class *P*; $L_{g,P}$ = number of '*l*' values for gene *g* assigned to class *P*; $H_{g,N}$ = number of '*h*' values for gene *g* assigned to class *N*; and $L_{g,N}$ = number of '*l*' values for gene *g* assigned to class *N*. As an example, these values

Formula (3), below, computes a rank for each gene that measures the power of the gene to distinguish between the two classes:

$$r_g = \left(H_{g,P} \times L_{g,N}\right) - \left(H_{g,N} \times L_{g,P}\right) \tag{3}$$

For a completely distinguishing gene where, all of its values for class *P* are '*h*', and all of its values for class *N* are '*l*', $H_{g,N} = L_{g,P} = 0$ and, r_g, takes its maximum *positive* value. In this case the gene is considered to be descriptive of (associated with) class *P*.

The gene remains completely distinguishing in the inverse case where, $H_{g,P} = L_{g,N} = 0$ and, r_g, takes the minimum *negative* value. In this case the gene is associated with class *N*. In other words the gene ranking formula encompasses and expresses a *polarity* characteristic that represents the descriptive power of the gene with respect to the present disease-state classes. So, ordering the list of positive and negative ranks in descending order may identify the most discriminant genes for class *P* and *N*, respectively. For example by considering and selecting genes just from the top of the two lists. Formula (3) could be considered as (and actually presents) a discrete

analogue of the respective signal-to-noise formula (used by various gene ranking and selection approaches, see for example [142]).

4.2.2.2 Gene-selection via Feature Elimination / Addition

Rank-ordering of the genes and selection for the top ranked genes does not solve the problem of 'how many genes' should be considered as the most discriminant. In most of the published gene-expression studies the researchers decide on an 'ad hoc' basis choosing a threshold cut-off value for this (i.e., [142]). Here we introduce a more careful and sound method that selects the most discriminant genes from the two rank-ordered lists. It consists of two processes.

- ***Grouping of genes.*** With this method we group genes that have similar ranking. First we estimate the value:

$$g = \frac{MaxRank - MinRank}{n-1} \tag{4}$$

MaxRank and *MinRank* are the maximum and minimum ranking of the genes respectively as they were computed from the previous step. As we have positive and negative ranking we have to estimate two *g* values: one for positive and one for negative ranking. Gene *i* is assigned to a group O_i according the formula:

$$O_i = \begin{cases} 1 & , i = 1, k \leftarrow 1 \\ k & , R_i - R_{i-1} \leq g \\ k+1 & , R_i - R_{i-1} > g, k \leftarrow k+1 \end{cases} \tag{5}$$

In this formula, R_i is the ranking of gene *i,* and *k* is an integer variable.

- ***Greedy gene-groups elimination.*** We are presented with the two vectors of groups of genes, $O_P = <O_{p;f}, O_{p;f-1} \ldots O_{p;1}>$and $O_N = <O_{n;f}, O_{n;f-1} \ldots O_{n;1}>$. Note that the beginning elements in the two vectors contain groups of genes that are less distinguishing between the two classes. In contrast, the ending elements contain genes that are most discriminant. So, it is rational to consider a procedure that eliminates groups from the beginning of the two vectors. We consider three situations: (i) deleting a group from O_P, (ii) deleting a group from O_N, and (iii) deleting a group from both O_P and O_N. In all cases, the accuracy of the remaining genes on the training samples is assessed. The accuracy is computed based on a specially devised *predictor* metric (presented in the next chapter). The accuracy figure and the respective list of remaining genes are recorded. The deletion that exhibits the highest accuracy is performed. The group-elimination process continues till all the groups in the two lists are considered. The list of remaining genes with the highest accuracy is selected as the final set of most discriminant genes.
- ***Greedy gene-groups addition***. Greedy feature selection could be implemented with its *dual* namely, *greedy gene-groups addition*. In this mode, groups of genes are added to an initially empty list until the prediction performance declines – the genes accumulated to that pointed compose the finally selected genes. In this initialization phase, with empty set of genes, the perspective performance is

considered to be the default one (i.e., the majority class is used for class prediction of all samples).

Note. In the context of MineGene both gene-groups elimination and addition methods are implemented. Moreover, the same methods are implemented for eliminating/adding just ***one gene at a time***, i.e., without forming groups of ranked genes.

4.2.3 Samples Class Prediction

The vision of functional genomics, at least for the human case, is the devise of diagnostic and prognostic kits for various diseases. With the utilization of microarray chip technology the target is to devise microarray *chip-based diagnostic* and *prognostic kits* dedicated to specific diseases. In the core of the process for devising such a kit are gene-selection methods, much in the sense presented above. Having in our disposal such a kit the question is how a new patient (i.e., its potential pathologic sample-tissue) is classified to a disease-state class or, how its prognosis is predicted.

Assume that the sample is presented as a vector of gene-expression values for the genes that are present in the diagnostic/prognostic kit. We introduce a novel matching procedure, and a respective metric, that predicts the class of a sample.

We assign the integer values '*1*' and '*-1*' to the respective discretised genes' expression-levels of the new sample (we have already mentioned that the gene expression values of an unseen sample are discretised according to the mid points computed during the training phase). The integer values '*1*' and '*-1*' stands for the '*h*' and '*l*' assignments, respectively, denoted with $sign(s_g)$. The matching formula (3), below, is used to predict the class of a sample *s*.

$$class(s) = \left[\sum_{g \in P} \left(sign(s_g) \times \frac{H_{g;\,P} - L_{g;\,P}}{|P|} \right) - \sum_{g \in N} \left(sign(s_g) \times \frac{H_{g;\,N} - L_{g;\,N}}{|N|} \right) \right] \quad (6)$$

In this formula, with $g \in P$ we denote all selected and positive ranked genes and respectively with $g \in N$ we denote all selected and negatively ranked genes. With |P| and |N| we denote the number of "Positive" and "Negative" train samples respectively. As with the gene-ranking formula (formula 3, above) formula (6) also encompasses a polarity characteristic. If the outcome of the formula is positive then the new sample is assigned to class *P*, and if it is negative then it assigned to class *N*. In addition, the strength with which the sample is predicted to belong to one of the two classes is also provided so that, *strong* (or, *weak*) predictions could be made. Take as an example the extreme case were $L_{g;P} = H_{g;N} = 0$ for all selected genes (i.e., all the genes have *'high'* values for all class *P* samples, and *'low'* values for all class *N* samples; in other words all selected genes are ideally associated with the respective classes). Then, in formula (6) the bracketed factor receives its maximum positive value which equals the total number of total selected genes, say T. Now, if the incoming unseen sample have *'high'* values (i.e., $sign(s_g) = 1$) for all genes associated with class *P*, and *'low'* values (i.e., $sign(s_g) = -1$) for all genes associated with class *N* (i.e., an ideal class *P* sample) then, formula 6 receives its maximum positive value which equals to -T. So, the sample is strongly predicted to belong to class *P*. All the above holds for the inverse case where, the incoming sample is an

ideal class *N* sample- the outcome of formula 6 will be *-2S*, and the sample will be strongly predicted to belong to class *N*. Under suitable assumptions (based on an analysis of all prediction figures) a 'weak' prediction could leave the sample *unclassified*.

4.2.4 Multi-domain Prediction Method

The two-class predictor not only uses the metric of entropy to decide which class should be assigned to an unclassified sample, but also produces a strength of prediction value according to the relevance of the unclassified sample to our train samples. This strength can be applied to tackle domains with more than two classes.

- Let *S* be an unclassified sample that belongs to a domain with *c* classes. We also assume that we have selected *g* genes to be our discriminant attributes. We apply the predictor described above subsequently for each class. That is, we estimate the prediction strength of *S* belonging to each one of the *c* classes. Finally we assign the sample *S* to the class that made the best prediction score. With this mode of operation (also implemented in MineGene) we are able to predict the class of samples in the presence of multi-classes – an operation of great value in the case of multi-disease (e.g., multi-cancer) domains.

4.3 Experimental Evaluation of the MineGene Gene-Selection Methodology

We applied the introduced gene-selection and samples classification methodology on eight real-world gene-expression domain studies that are pioneers in their fields. A total of six biomedical domains were investigated and respective tasks were posted (for two domains, HBC and CNS, two different tasks are posted). Below the respective reference studies and tasks, with which we compare our gene selection method, are listed.

- ***LEUK*** (Leukemia; Ref. [142]) – to distinguish between two leukemia classes, *ALL* and *AML*;
- ***BRCA*** (Breast Cancer; ref. [143]) – to distinguish between two classes, patients with *no metastasis* in at-least five years and patients with *metastasis* within five years;
- ***COLON*** (Colon Cancer; Ref. [144] for original study, and [146] for the comparison reference) – the task is to distinguish between *normal* and *tumor* samples
- ***LYMPH*** (Lymphoma; Ref. [146]) – to distinguish between two lymphoma-characteristic classes, *GCB* and *AB* (types of cells);
- ***HBC*** (BRCA1; Ref. [147]) – to distinguish between *BRCA1* and *not-BRCA1* mutated samples;
- ***HBC*** (BRCA2; Ref. [147]) – the same as the previous domain study but with the task of distinguishing between *BRCA2* and *not-BRCA2* mutated samples;
- ***CNS*** (Meduloblastoma; Ref. [148]) – to distinguish between two types of meduloblastoma brain tumours, *Classic* and *Desmoplastic*;
- ***CNS*** (Treat.Outcome; Ref. [148]) – the same as the previous domain but with the task of distinguishing between two treatment outcomes for patients with meduloblastoma, *Survivors* and *Failures*.

In table 2 the specifics (e.g. reference study, number of genes, classes, etc) of the above domains are presented.

Table 2. *Experimental domain studies: Comparison reference studies and respective datasets.*

Study #	Study Name (Task)	Study Reference	Classes	#Genes	Training Samples
1	**LEUK**	[142]	{ALL , AML}	7129	38 {27,11}
2	**BRCA**	[143]	{RELAPSE , NON-RELAPSE)}	24481	78 {34,44}
3	**COLON**	[144]]	{TUMOUR , NORMAL}	2000	62 {40,22}
4	**LYMPH**	[146]	{GCB , AB}	4026	47 {24,23}
5	**HBC (BRCA1)**	[147]	{BRCA1 , notBRCA1}	5361	22 {7,15}
6	**HBC (BRCA2)**	[147]	{BRCA2 , notBRCA2}	5361	22 {8,14}
7	**CNS (Medulloblastoma)**	[148]	{CLASSIC , DESMOPLASTIC}	7129	60 {9,25}
8	**CNS (Treatment Outcome)**	[148]	{SURVIVORS , FAILURES}	7129	60 {39,21}

4.3.1 Results and Discussion

Table 3, summarizes the results of applying the introduced MineGene gene-selection and sample classification/prediction method. The bold figures indicate superior performance with respect to the reference study, and: (a) to the number of selected genes (i.e., less number of genes is considered as superior), and (b) to accuracy assessment results.

Table 3. *Comparison results assessed on the available training samples: MineGene vs. stuffy reference results. Bold figures indicate superior performance (better accuracy and less number of selected genes). #G: number of selected genes; Acc%: accuracy figure (%); Ref.: the number of selected genes and the reported (in the original reference paper) accuracy; MineGene.a: number of selected genes and accuracy figures for the MineGene gene addition method; GeneMin.d: the same as previous but for the MineGene gene deletion method; Ref./PRED: using the reported (reference) genes with MineGene's predictor; Ref./MineGene.a: number of selected genes and accuracy figures when MineGene.a (both selection of genes and prediction) is applied just on the reported genes; Ref./MineGene.d: the same as previous but when MineGene.d was applied (figures in bold shows superiority over the reference study results).*

Study #	Study Name (Task)	MineGene.a/d				STUDY						
		MineGene.a		MineGene.d		Ref.		Ref./PRED	Ref./MineGene.a		Ref./MineGene.d	
		#G	Acc%	#G	Acc%	#G	Acc%	Acc%	#G	Acc%	#G	Acc%
1	LEUK	**1**	**100.0**	**4**	**100.0**	50	94.7	81.6	**5**	**94.7**	**13**	**97.4**
2	BRCA	**33**	**97.4**	**34**	**97.4**	70	83.3	71.8	**17**	**87.2**	**63**	**84.6**
3	COLON	127	**100.0**	26	**100.0**	**10**	**100.0**	90.0	**6**	92.5	**6**	92.5
4	LYMPH	**4**	**100.0**	**4**	**100.0**	50	97.1	**100.0**	**4**	**100.0**	**4**	**100.0**
5	HBC (BRCA1)	10	**100.0**	10	**100.0**	9	95.5	95.5	**5**	**100.0**	**5**	**100.0**
6	HBC (BRCA2)	**3**	**100.0**	**5**	**100.0**	11	81.8	**100.0**	**4**	**100.0**	**6**	**100.0**
7	CNS (Medul/stoma)	**21**	**100.0**	**21**	**100.0**	140	97.1	97.1	**17**	**100.0**	**11**	97.1
8	CNS (Treatment Outcome)	**10**	**95.0**	**32**	**93.3**	100	78.3	**91.7**	**19**	**95.0**	**39**	**96.7**
	MEAN	**26**	**99.1**	**17**	**98.8**	55	91.0	90.9	**10**	**96.2**	**18**	**96.0**

- ***Accuracy assessment.*** As it can be observed, the introduced gene-selection methodology outperforms, in most of the cases, the ones in the comparison-references. At an average, the accuracy achieved with MineGene is 99.1%, and 98.8%, when using the gene addition and deletion approaches, respectively. These figures should be compared with the reported (in the original study reference publication) accuracy figure of 91.0% - a statistically significant difference on the P>99% level, for both MineGene.a and MineGene.d, applying a *one-tail* t-Test on the accuracy figures over all domains.

  The results show the reliability of the introduced MineGene gene-selection and sample classification/prediction methodology. The high performance could be attributed not only to the overall gene-selection approach (i.e., discretisation, gene-ranking and gene-selection) but also to the introduced prediction metric and methodology. In particular when the reported genes where used for prediction of samples' class an average accuracy figure of 90.9 % was achieved. This figure is comparable with the 91.0% average reported accuracy figure, also confirmed with the observation of no statistically significance difference between the respective accuracy figures (even for the P>90% level when the same as above statistical test was applied).

- ***Number of genes.*** Furthermore, MineGene results in a significant smaller number of selected genes, an average of 26, and 17 for the MineGene gene addition and deletion approaches, respectively, compared with an average of 55 reported genes for the reference studies. A statistically significant difference was observed on the P>95% level, applying a *two-tail* t-Test statistical test for the respective numbers of selected genes (i.e., 26 and 17 vs. 55). This result is quite satisfactory because a small number of disease associated genes gives the opportunity for more complete and better biological interpretation (e.g., for the involved disease-related biochemical pathways).

  Furthermore, using just the reported genes as a starting point for the MineGene method we were able to achieve even better results. In particular, high average accuracy figures of 96.2%, and 96.0% were achieved when MineGene was applied with the gene addition, and deletion approaches, respectively (a statistical significance difference on the P>90% level for a one-tail t-Test). In this experimental mode we were able to find even less number of discriminatory genes – 10, and 18 for the gene addition and deletion approaches, respectively (with a statistical significance difference on the P>99% level for a one-tail t-Test).

4.4 Future R&D work for Gene-Selection

The future research agenda includes: (a) further experimentation with other gene-expression profiling domains, especially multi-class (more than two) domains, (b) biological interpretation of the results (e.g., how many of the selected genes are common in our results and the original comparison references), and (c) inclusion of the gene-selection and samples classification methodology in an Integrated Clinico-Genomics Environment to ease decision making in the genomic medicine context [139].

Whether we use supervised or unsupervised expression profile analysis, they are just the first steps in expression data analysis. It is a long way from finding gene clusters to finding functional roles of the respective genes, and moreover, understanding the underlying biological processes. A natural step downstream of expression profile clustering is the usage of putative promoter sequences of similarly expressed genes for finding regulatory sequence elements in genomes. This is easier from yeast, since typically yeast promoters are relatively close to ORFs.

5. Discovery of Co-Regulated Genes: A Clustering Approach

The goal of clustering is to group together objects (genes or samples) with *similar properties*. This can also be viewed as the reduction of the dimensionality of the system or, the discovery of "*structure in the data*". By comparing gene-expression profiles, and forming *clusters*, we can hypothesize that the respective genes are co-regulated and possibly functionally related.

- In this setting, clustering serves for the discovery and identification of potential genes' ***function***. The discovery of genes' function may help to the identification of genes being involved in particular ***molecular pathways***, and by though ease the modelling and exploration of ***metabolic pathways*** (i.e., *metabolomics*).
- Moreover, clustering of genes may reveal ***gene-families***, i.e., ***metagenes***, and their potential *linkage with combined clinical features* – a task which is *too-difficult* to be achieved when we are confronted with the huge number of available genes (~25000-30000 for the human case).

The current thesis introduces a novel ***graph theoretic clustering*** (GTC) approach. The approach is based on a graph-based arrangement of the input objects (genes in our case). With a careful and iterative partitioning of the graph's *minimum spanning tree* (MST) it results into a hierarchical clustering of the input objects.

5.1 State-of-the-art Approaches and Utility of Clustering Microarray Data

The goal of clustering is to group together object (genes or samples) with similar properties. Many clustering algorithms have been applied to analyze expression data. The hierarchical [160] and K-mean clustering algorithms [178], [20], [179] as well as self-organizing maps [180] have all been used for clustering expression profiles.

Clustering of expression profiles has been used for grouping genes as well as samples. The clustering of genes for finding co-regulated and functionally related groups is particularly interesting in the cases when we have complete sets of an organism's genes. DeRisi et al. [181] used a DNA array containing a complete set of yeast genes to study to dauxic shift time course. They selected small groups of genes with similar expression profiles and showed that these genes are functionally related and contain relevant transcription factor binding sites upstream of their ORFs. More systematic studies of this dataset for regulatory elements were done by Brazma et al. [166] and Helden et al. [182].

Later more expression studies of yeast under various conditions were carried out, including sporulation [183], cell cycle [184] and yeast gene regulation machinery [185]. Clustering has been applied to the obtained gene expression matrices, and groups of functionally related and co-regulated genes have been revealed. Tavazoie et al. [179] clustered expression profiles of 3000 most variable yeast genes during the cell cycle into 30 clusters by the K-means algorithm. They found that for half of these clusters, strong sequence patterns are present in the gene upstream sequence. Note that expression profiles of cell cycle-dependent genes are periodic and Fourier analysis has been used to discover these genes [184].

Eisen et al. [160] have developed a hierarchical clustering-based algorithm and visualization software package, which is currently one of the most frequent used tools for expression profile clustering and data visualisation. They applied their software to

gene expression matrices obtained by combining 80 different yeast samples (experimental conditions) studied in various hybridization experiments at Stanford University.

Gene expression profile clustering does not necessarily require the full genome. For instance Iyer et al [186] studied 8600 genes in human fibroblasts and obtained 10 distinct gene clusters each associated with genes with particular functional roles, such as signal transduction, coagulation, hemostasis, inflammation etc.

A simple method of finding sets of interesting genes is comparing expression profiles of two or more samples for differentially expressed genes. For instance, Lee et al. [187] used this method to find genes that are differentially expressed in skeletal muscle of adult (5 months) and old (30 months) mice. Of over 6347 mouse genes surveyed by a microarray, 58 displayed a greater than two-fold increase, whereas 55 displayed a greater that two-fold decrease in expression in the skeletal muscles of the old mice.

Ben-Dor et al. [188] applied a new clustering algorithm for classification of colon and ovarian cancer data sets. They used unsupervised clustering to find a hierarchical structure in the expression profile space, and supervised learning to find the best threshold to correlate the clustering structure with the known cancer classes.

Hierarchical clustering has also been used for sample clustering. An interesting application of this approach is the clustering of tumours to find new possible tumour subclasses. Alizadeh et al. [146], applied this approach where diffuse large B-cell lymphoma (DLBCL) was studied using 96 samples of normal and malignant lymphocytes. Applying a hierarchical clustering algorithm to these samples they showed that there is diversity in gene expression among the tumours of DLBCL patients forming two distinct clusters. These two groups correlated well with patient survival rates, thus confirming that the clusters are meaningful.

Sample clustering has been combined with gene clustering to identify which genes are the most important for sample clustering [146], [144]. Alon et al. [144] have applied a partitioning based clustering algorithm to study 6500 genes of 40 tumor and 22 normal colon tissues for clustering both genes and samples. They call this method two-way clustering.

Another fact that indicates the significance of the clustering methods can be found in gene regulatory networks, where we try to identify the role of every functioning part of a gene by doing something like “reverse engineering”. Based on the hypothesis that genes that have similar expression profiles (i.e. similar rows in the gene expression matrix) should also have similar regulation mechanisms as there must be a reason why their expression is similar under a variety of conditions. Therefore, if we cluster the genes in such clusters, some of these sets of sequences may contain a ‘signal’ as a specific sequence pattern such as a particular substring, which is relevant to regulation of these genes.

5.2 A Graph Theoretic Clustering (GTC)

In this chapter we present a novel Graph Theoretic Clustering (GTC) approach on clustering of microarray gene expression profile data. The approach is based on the arrangement of the genes in a weighted graph, the construction of the graph’s Minimum Spanning Tree (MST), and an algorithm that recursively partitions the tree.

5.2.1 Related approaches and utility of GTC clustering approach

MST-based clustering is not a new idea. It was first introduced by Zahn [189] and Page [190]. Recently a similar approach that follows a different partitioning strategy was also introduced and applied on gene-expression profiling tasks [191]; the method is implemented in the core of the EXCAVATOR gene-expression analysis system [192]. These approaches follow a 'one-shot' MST partition strategy with the identification of 'weak' (or, 'long') MST edges, which are then cut. Because of their one-shot partitioning strategy these methods could not identify special relations in the data as for example the potential of a hierarchical organization. In addition, all approaches demand the presetting of the number of desired clusters. In most cases such a demand is problematic, especially in exploratory data analysis where, the analyst possesses no hints about the potential number of clusters. For the approach in [191] an estimate for the optimal number of clusters is computed in advance, a pre-processing step of high computational cost.

Moreover, GTC exploits a 'hybrid' characteristic. Assuming that the assignment of genes to classes is known in advance, or we have an external source of information that can estimate an arbitrary form of distance between two genes then, several metrics and distances can be used to utilize information that comes from this external (to the expression-based description of the genes) modality. The clustering is to be performed an a (potentially) different distance-based arrangement of the genes, and the final hierarchical clustering outcome reflects both: (a) the expression-based description of the genes and (b) their class assignments. So, conjectures made from one source of information may be used to confirm (or, reject) conjectures from the other, and vice versa. In this setting, pre-established domain-knowledge is utilized in order to discover regularities and confirm/reject hypotheses. In that sense, GTC presents a 'knowledgeable' exploratory data analysis approach. This is in contrast to other MST-based clustering approaches where, the computation of distances between objects relies solely on the expression-based description of the objects and the corresponding 'geometric' arrangement of them. In this mode clustering is not coupled with background domain knowledge, a crucial source of information in order to decide where to cut the MST (especially for 'borderline' cases).

With GTC there is no need to specify the number of clusters in advance (a prerequisite of other clustering approaches such as k-means [20]). In contrast, a 'termination' condition, implemented with an information-theoretic formula, is applied on each of the nodes of the growing cluster-tree and decides to stop or, to further expand the tree at that node. A special feature of GTC is the combination of different information sources in order to compute the distance between the input objects (genes). Domain background knowledge can be utilized in order to compute distances between objects and arrange them in a weighted graph. Then iterative partitioning of the respective MST is done with reference to the original feature-based description of data. This hybrid characteristic makes the whole data analysis process more 'knowledgeable' in the sense that established domain knowledge guides the clustering process. The final result is a hierarchical clustering-tree organization of the input gene expression profiles. We focus on the discovery of indicative and descriptive patterns in order to 'uncover' hidden relations and yield insights on the order of spatial maps of genome, providing profiling rules that possibly reveals its functional structure and selective transcription.

5.2.1 Minimum Spanning Tree Construction

With the microarray gene expression matrix in our disposal we compute the distances of all gene expression profiles. The distances between all the genes expressions profiles can be a simple (i.e. Euclidean, Manhattan) distance or something more domain specific suitable to reveal certain data regularities (i.e. Pearson, Mahalanobis). It also can be, as we have discussed, a complete arbitrary, external source of information.

The next step is to form a fully connected weighted graph, with the genes as nodes and computed distances as edge-weights. In order for this graph to be formed all combinations of gene distances must be computed. If we have *n* nodes (genes) then the graph will have $\frac{1}{2}(n-1)n$ edge-weights (fully connected, figure 19)

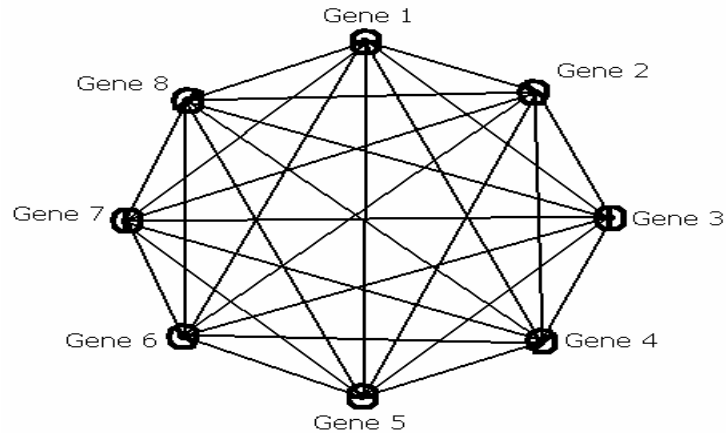


Figure 19. Connected *graph: Each node is a gene; the weight of each edge is not shown*

Given a set *E* of *n* genes, the minimum spanning tree of the fully-connected weighted graph of the objects is constructed. The formed MST contains exactly *n-1* edges. In the current GTC implementation we used Prim's [193], Kruskal's [194] and Round Robin [195] methods for the construction of the MST. A basic characteristic of the MST is that it reserves the shortest distance between the genes (figure 20). This guarantees that objects lying in 'close areas' of the tree exhibit low distances. So finding the 'right' cuts of the tree could result in a reliable grouping of the genes.

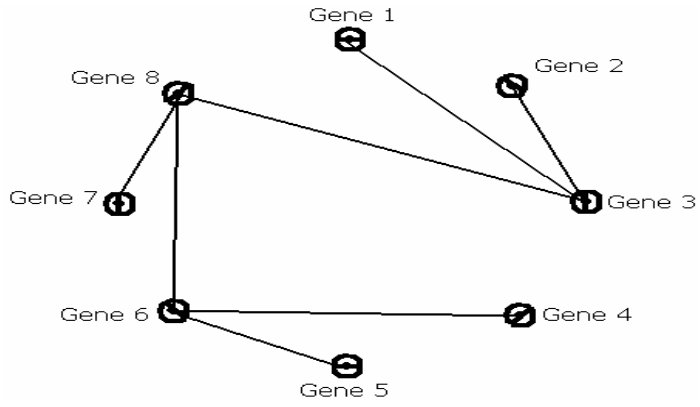


Figure 20. *The Minimum Spanning Tree of the graph in figure 19 (for given weights of links).*

5.2.2 Iterative MST partition

Iterative MST partition is implemented within the following three steps.

Step1: Binary splitting. At each node (i.e., sub-cluster) in the so-far formed hierarchical tree, each of the edges in the corresponding node's sub-MST is cut. With each cut a binary split of the genes is formed. If the current node includes *n* genes then *n-1* such splits are formed (figure 21). The two sub-clusters, formed by the binary split, plus the clusters formers so far (excluding the current node) compose a potential partition.

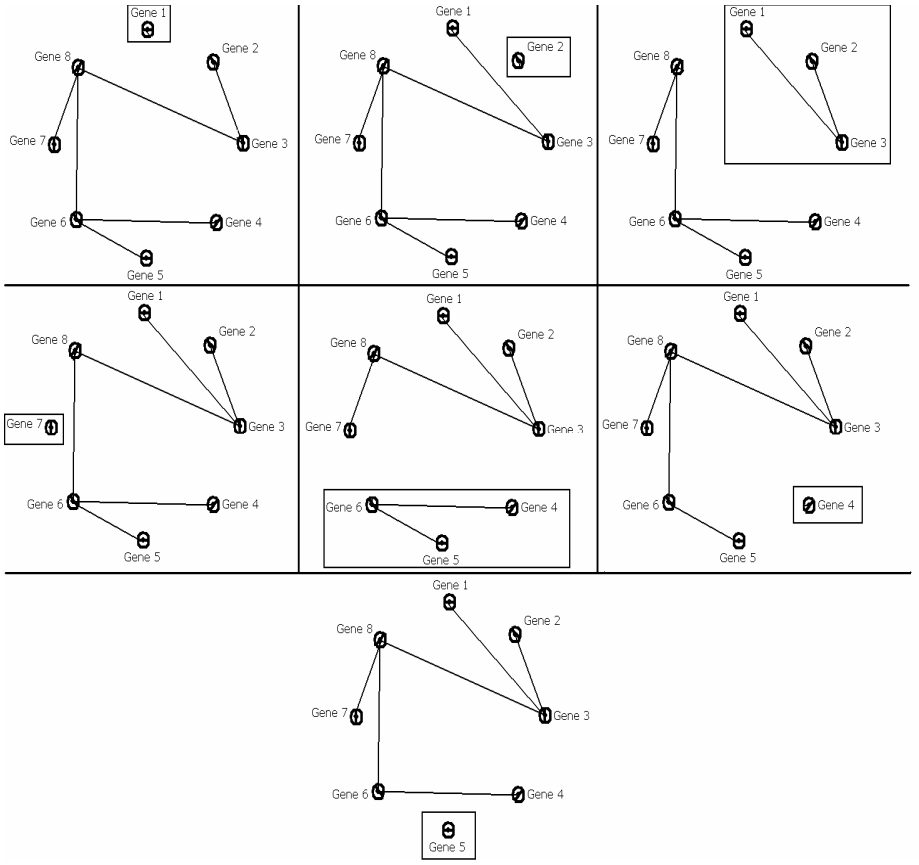


Figure 21. *Binary splitting of a MST.*

Step 2: Best Split. For each binary split we compute a category utility (CU) that indicates the division ability of the split. The more compact the clusters formed the higher the CU. As the expression profile data is numeric, we assume that the probabilities for numeric attributes have a normal distribution, and we use the height of the normal curve as the probability of a particular attribute value. The following formula shows this derivation process [196]:

$$\sum_j P(A_i = V_{ij})^2 \Leftrightarrow \int \frac{1}{\sigma^2 2\pi} e^{-\left(\frac{x-\mu}{\sigma}\right)^2} dx = \frac{1}{\sigma 4\sqrt{\pi}} \qquad (7)$$

The transformed evaluation function is [197]:

$$CU_{numeric} = \frac{\left(\sum_k P(C_k)\sum_i \frac{1}{\sigma_{ik}}\right) - \sum_i \frac{1}{\sigma_{iP}}}{4K\sqrt{\pi}} \quad (8)$$

Where K is the number of clusters formed so far, σ_{ik} is the standard deviation for samples i in class k, and σ_{iP} is the standard deviation for attribute i of all the genes participating in the clustering. The one that exhibits the highest CU is selected as the best partition of genes in the current node.

Step 3: Iteration and termination criterion. Each new cutting point found on the tree, divides the tree in two sub-trees, let the first sub-tree be the *left* and the second the *right*. The best cut of these two trees is found as described in steps 1 and 2. In order to decide what will be the new cut, four potentials have to be examined. The first is that none cut should be considered, thus none new cluster will be formed and the algorithm must terminate. The second is that the best cut in the *left* sub-tree should divide the *left* tree into 2 new clusters while the *right* tree should not be examined any more. The third potential is symmetric to the second, thus the *right* tree should be divided and the *left* to be remained stable. The forth potential is that both cuttings should be considered and both *left* and *right* tree should be divided. In order to decide what potential is the proper one we estimate the CU of each one and select the one that exhibits the higher value. Then for each new division decided we iterate steps 1 through 3 (see figure 22).

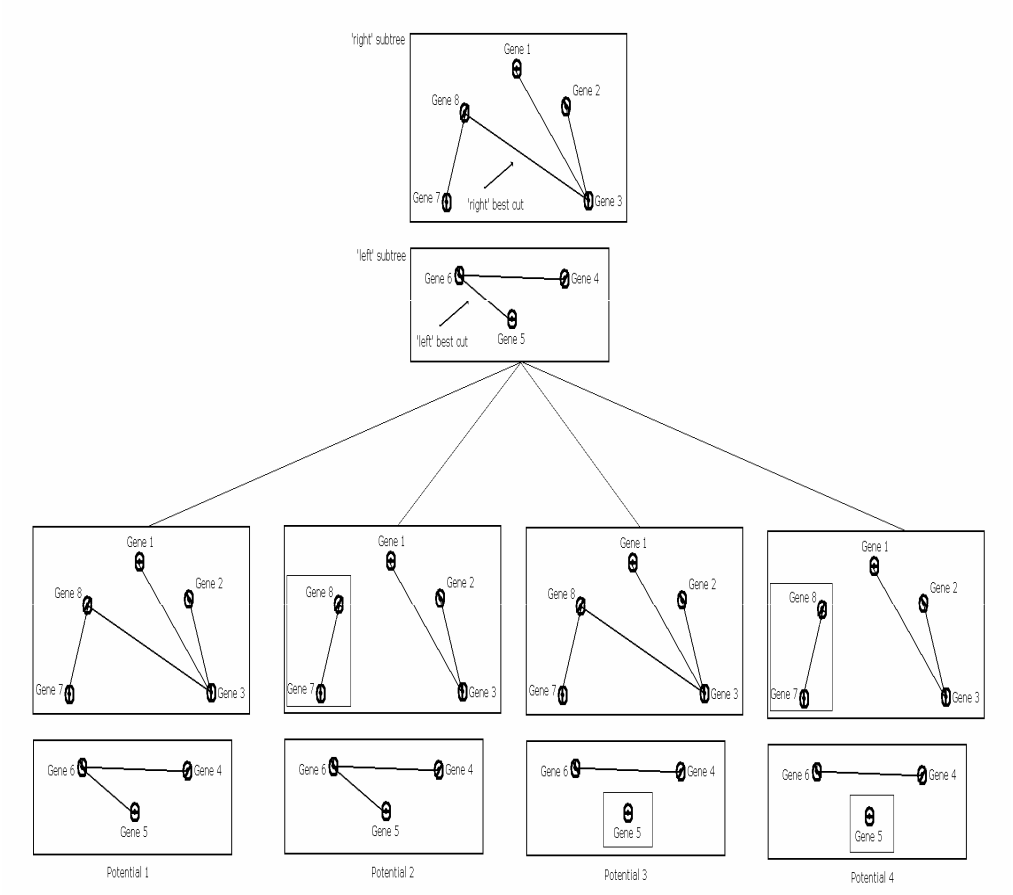


Figure 22. *Four potentials of a partitioning step.*

The final outcome is a hierarchical clustering tree where (by default) the termination nodes are the final clusters. After visual inspection of the hierarchical tree the user may decide to use higher levels of the tree as the final clustering. Note that there is no need to determine the number of clusters in advance – a task left to the node growing/termination criterion (step 3).

5.2.3 Time complexity of GTC: Preliminary Assesment

The core of GTC (i.e., the MST recursively partitioning) time-complexity depends: (i) on the complexity of computing the category utility indices and (ii) on the depth of the resulted clustering tree. Denote with *F*, the number of features (samples) and *n*, the total number of input objects. The category utility computation needs a time linear to the total number of the features, $\approx \mathrm{O}(F)$.

In the worst case the maximum depth of the tree is *n-1*. That is, at the zero level (i.e., all genes in one group) the resulted sub-clusters have *1* and *n-1* objects, respectively. The sub-clusters are formed after performing a total of *n-1* CU computations (i.e., edge cut or, splits of the corresponding MST tree. At the second level the clusters with 1 and n-2 objects, respectively, after performing a total of *n-2* CU computations. At the last level, *n-1*, there are *n-(n-2)* genes, and a total of *n-(n-2)-1=1* CU computations are to be performed. So he total number of CU computations is equal to *1+2+…+(n-1)=n(n-1)/2.* As a result, and for the worst case, the GTC algorithm exhibits a quadratic to the total number of genes, and linear to the total number of samples, time-complexity, i.e., $\approx O(n^2 \times F)$.

The quadratic complexity figure is in accordance to hierarchical clustering approaches that use dynamic closest pairing techniques [198], and with k-means approaches when the preset number of clusters is equal to the total number of input objects. The time complexities of the MST algorithms are: Prim's $\sim O(n \log_2 F)$, Kruskal's $\sim O(n \log F + Fa(n))$ and Round Robin $\sim O(n \log\log F)$. As we can see all of them are significantly faster than the main GTC algorithms, thus cannot be considered in time complexity estimations.

In all the conducted experiments, and for datasets with ~1000-27000 genes and ~78-100 samples, the real execution time of the C++ based GTC implementation ranges from ~2 to ~30min (on a 3.2MHz, 1Gb RAM PC).

5.2.4 Coping with Time Complexity: Keep 'Significant' Weighted Links

One of the main bottlenecks of the algorithm is the distance calculation. The time complexity (and space complexity) of calculating all distances of *n* genes with *F* samples is $\sim \Theta(F \times n^2)$. Especially when dealing with gene expression matrices the number of input objects may reach the value of 30,000 (35,000 numbers is the estimation number of human genes), thus the time and space requirements of the algorithm can reach the order of 10^{11}. Even though this complexity can be arranged by contemporary modern computers in the field of time, it is very hard to be arranged in the field of space. In order to overcome this bottleneck we introduce a heuristic that reduces significantly the order of the computed distances.

We assume that the maximum degree of computed MST's nodes is a value less than a constant value, let *t*. This hypothesis comes from the belief, that the data has a

minimum sparseness. Even though gene expression data have many irregularities, we can safely assume that a cluster can have a maximum compactness. Thus a MST of a fully connected graph cannot have a node with degree greater than *t*. As a consequence it is adequate to compute the *t* minimum distances of each node. This reduces the space complexity to $\sim\Theta(F \times t \times n)$ even though it increases the time complexity as the burden of sorting the distances of each node has been added.

The resulted graph will not be fully connected, but the produced MST will be exactly the same if the *t* value is not too small. According to our implementation a value of *t* close to 1% of the number of input objects (genes) proved to be a rational value.

5.3 Experimental Evaluation of GTC on Gene-Expression Data Clustering

We utilized GTC on an indicative gene expression profiling domain namely, large scale gene expression profiling of central nervous system development, referred as the *Wen* case-study [199]. The respective case-study present the mRNA expression levels of 112 genes during rat central nervous system development (cervical spinal cord); assignment of the 112 genes to four main functional classes- spitted further to fourteen class-values is also provided.

Utilizing a special devised distance measure, the VDM metric (see below), in the course of GTC five clusters were induced that exhibit, not only similar expression profiles but similar, more-or-less, functions as well. The natural interpretation of the induced clusters and their correspondence to the respective Wen 'w'aves are: EARLY / w1; MID-LATE / w2; MID / w3; LATE / w4; and CONSTANT / w5. Figure 23, below, shows the representative profiles for each of the induced clusters (the plotted patterns present the developmental-stage means over all genes in the respective cluster).

Note. In the current MineGene implementation the Euclidean distance is implemented – the VDM metric was utilised off-line (of the MineGene system) and the results were saved in a file (appropriately formated to be read from MineGene).

5.3.1 Results and Discussion

The result shows that the presented clustering approach is well-formed and reliable producing similar results with the standard joining-neighboring clustering approaches (followed by Wen). Moreover, for all functional classes GTC/VDM exhibits lower diversity indices figures; compared with Wen's clustering a significance difference was observed on the P>99% level. So, the GTC/VDM clustering approach induces more 'compact', with respect to the genes' functions, clusters. Furthermore, in hierarchical clustering approaches it is difficult to identify the 'borderline' patterns, i.e., genes with expression profiles that lie between two or, more clusters. This is the situation with the w2/c2112 and w3/c2111 clusters. In Wen clustering there are some genes that are assigned to cluster w2, even if their expression patterns fits more-or-less to the w3/c2111 pattern. The GTC/VDM clustering approach remedies this, and groups the genes within cluster w3/c2111. A special case of 'bordeline' cases are the 'unclassified' ones – some genes assigned to the 'neuro_glial_markers' function remain unclassified in the Wen case study (the 'other' pattern in Wen's terminology). With CTC/VDM most of these genes are assigned to cluster w3/c2111 in which, most of the genes comes from the 'neuro_glial_markers' function. So, with the utilization of background-knowledge (i.e., knowledge about the function of genes) it is possible to solve the 'borderline' problem, and make the interpretation of the final clustering result more natural.

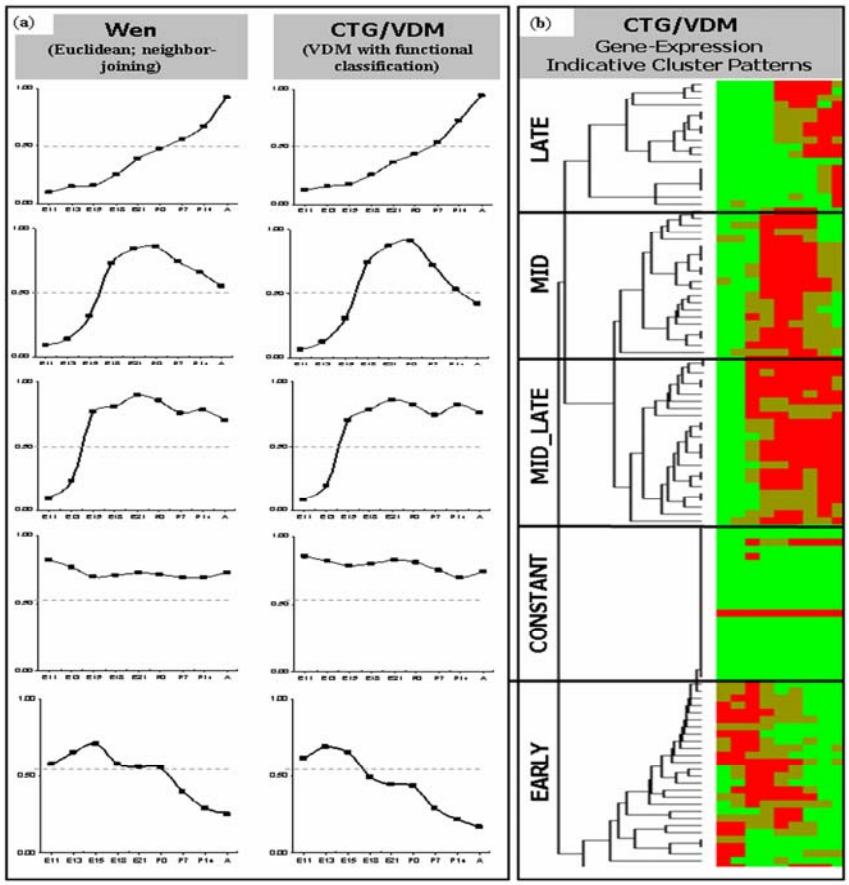


Figure 23. *Plots of the clusters' mean expression level (representative patterns) for Wen and CTG/VDM clustering.*

Value Difference Metric (VDM): A Knowledgeable Distance Measure. VDM combines information about the input objects that originates from different modalities. For example, the a-priori assignment of genes to specific functional classes could be utilized. The VDM metric, given by the formula below, takes into account this information [200].

$$\mathbf{VDM}_a(V_a = x, V_a = y) = \sum_{c=1}^{C} \left| \frac{N_{a;\,x;\,c}}{N_{a;\,x}} - \frac{N_{a;\,y;\,c}}{N_{a;\,y}} \right|^2 \qquad (9)$$

where, $V_a=x$: x is the value of feature a; $N_{a;x}$: the number of objects with value x for feature a; $N_{a;x;c}$: the number of class c objects with value x for feature a; and c the total number of classes. Using VDM we may conclude into a distance arrangement of the objects that differs from the one that results when the used distance-metric does not utilize objects' class information. So, the final hierarchical clustering outcome will confront not only to the distance between the feature-based (i.e., gene expression values) description of the objects but to their class resemblance as well. As the assignment of classes to objects reflect to some form of established domain knowledge the whole clustering operation becomes more 'knowledgeable'.

5.4 Future R&D Work for Clustering Microarray Data

The GTC clustering methodology is currently being tested on various domains, e.g., economic time-series data [201], mapping regional brain development [202]. The approach provides a framework where several distance metrics and category utilities can be applied and assessed. Thus, we need to expand our research on the direction to locate the most suitable distance metrics and category utilities for various research domains. In the field of microarray gene expression data we have to consider metrics that are resistant to erroneous or non-available data.

GTC methods are ideal for visualisation of inner data relations either during the method's process or the method's outcomes after termination. Existing visualisation techniques and available software does not provide visualisation in a large zoomed-out scale suitable for gene expression domains where the number of visualised objects usually exceeds the 30,000 nodes. In order to visualise and designate the inner gene relations stemming from GTC methods we have to introduce novel visualisation algorithms.

Finally, as proved one bottleneck of the algorithm, are the distances computations. Although we presented a heuristic that merely overcomes this problem, the need for advanced, sophisticated distance computation heuristics is still an open issue.

6. The MineGene System: Implementation Issues

In an integrated clinico-genomic environment we want to utilize a general-purpose machine learning tool to serve as an application platform for gene selection and clustering algorithms. This tool, names MineGene is a collection of Machine Learning algorithms and heuristics for intelligent processing of gene expression data produced by DNA Microarray experiments. Its main purpose is to mine into vast and redundant documents for information regarding the ability of certain genes to discriminate between different sample states. Similar tools are GeneSpring [203] and MolMine [204]. MineGene, is designed and implemented to be suited as a plug-in in a gene expression database. With MineGene we give the ability to a gene expression database apart from storing, retrieving, sharing and querying of the data, to infer fundamental conclusions about the inner regularities, descriptive ability and possible relations of the data stored.

The majority of the studies performed on gene expression data analysis follow a 'one-way' approach, thus they apply only one or a very limited set of algorithms. Even when a study is composed by many parts, each responsible for a specific aspect of the process, it is not possible to apply and test various algorithms for this aspect and infer invaluable conclusions not only for the data, but for the application spectrum of an algorithm as well. Moreover, even when we want to test a single algorithm it is desirable to have an environment capable to perform multiple runs with different inputs and parameters.

Judging from studies recently published, there is not yet any standard method for microarray gene expression data analysis but some general guidelines that recently have started to be formed. These guidelines represent a sequencing procedure that starts after data acquisition and ends to the construction of a predictor or a clustering mechanism depending if we are performing supervised or unsupervised data analysis (figure 24).

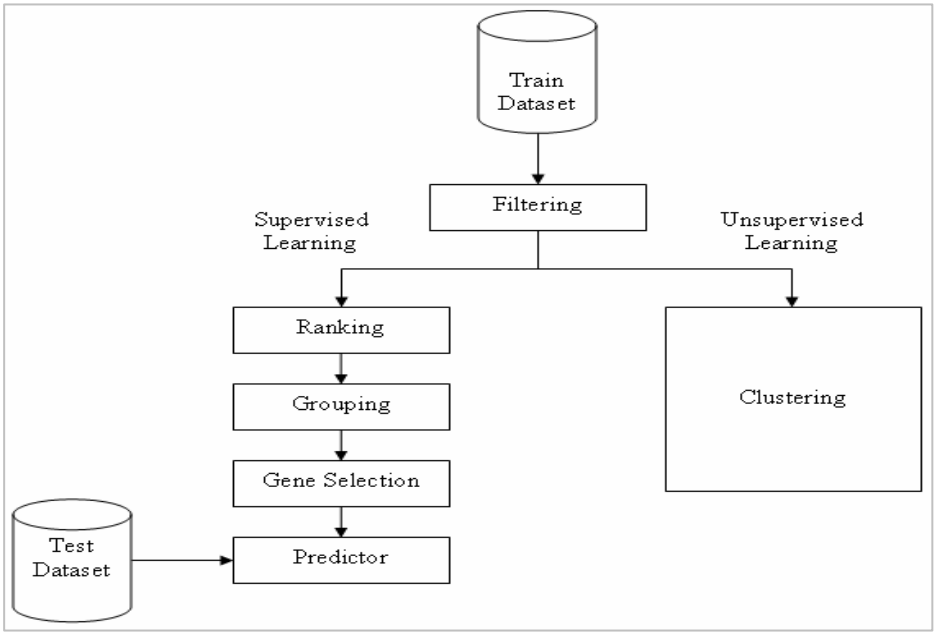


Figure 24. *Procedural tasks for gene expression data analysis.*

6.1 The supervised data analysis pathway

For supervised data analysis these guidelines form the following sequencing procedure:

- *Filtering.* Filtering is the first task of the procedure and the only that accesses the primary train dataset. Filtering can be considered as a preprocessing of primary data. With filtering we eliminate the number of further studied gene expression profiles according to a preferred criterion. The main reason to do this is to simplify the following tasks by providing them less data and to reduce the dimensionality of the problem. Usually the data filtered does not contain any significant information for gene expression regulations, namely filtered data do not significantly regulate among different sample classes. Some filtered methods include several hypothesis testing metrics as Wilcoxon rank-sum test and t-test.
- *Ranking.* With ranking we tag each gene with a value indicative of its descriptive ability. The higher the ranking the better the ability of the gene to discriminate between different sample classes. Some ranking methods include Pearson's correlation coefficients, standard deviation. The method proposed in chapter 4.2.2.1 is a ranking method.
- *Grouping.* Usually it is undesirable to manage each gene as a unique feature. The main reason for this is that according to previous step, some genes may exhibit a similar ranking, thus they should be treated as a group of genes. Moreover, treating each gene as a unique feature is sometimes an expensive computational task. Grouping allows as to reduce complexity and to emerge some physical correlation of the genes.
- *Gene Selection.* The next step is to select the most suitable genes that according to our methods can discriminate the samples among two or more categories. These genes must have been regulated differently in the two classes of samples. Most of the studies select an 'ad hoc' number of best ranked genes, but some algorithmic approaches exist as well. One of them is proposed in chapter 4.2.2.2.
- *Predictor.* The final step is to build the predictor. Here the genes selected from the previous task are chosen to act as attributes with continuous attribute values. Then each sample in the testing dataset is processed by the learning method selected here and assigned to a class. This is the only task where the test dataset is needed. Some famous learning methods include SVM, K-NN, K-means, as well as the one proposed on chapter 4.2.3.

6.1.1 Validation of Gene-Selection results: The Leave One Out Cross Validation (LOOCV) procedure

In cases where test dataset is not available, or we want to assess the predictive capacity of the train data we can use the Leave One Out Cross Validation (LOOCV) [163] method. During this method we take one sample from the train dataset. Then we perform all the algorithms described above, and then use the taken sample, as a test dataset. This process is done iteratively for all train samples. The ration of the samples predicted successful by the predictor reflects the predictive capacity of the train data. In the case than we have absence of test data we may consider as best discrimant the genes that participated more times in the selected genes set during a LOOCV procedure. LOOCV is an essential validation method that can estimate the value of our learning method and/or the predictive ability of our data.

6.1.2 Unsupervised data analysis pathway

For unsupervised data analysis these guidelines form the following two step procedure:

- *Filtering*. Filtering is exactly the same task as in supervised processes. Preprocessing of the data is still a very important task.
- *Clustering*. The clustering task is a generic unsupervised grouping method. Clustering significance and methods have been surveyed in chapter 5.

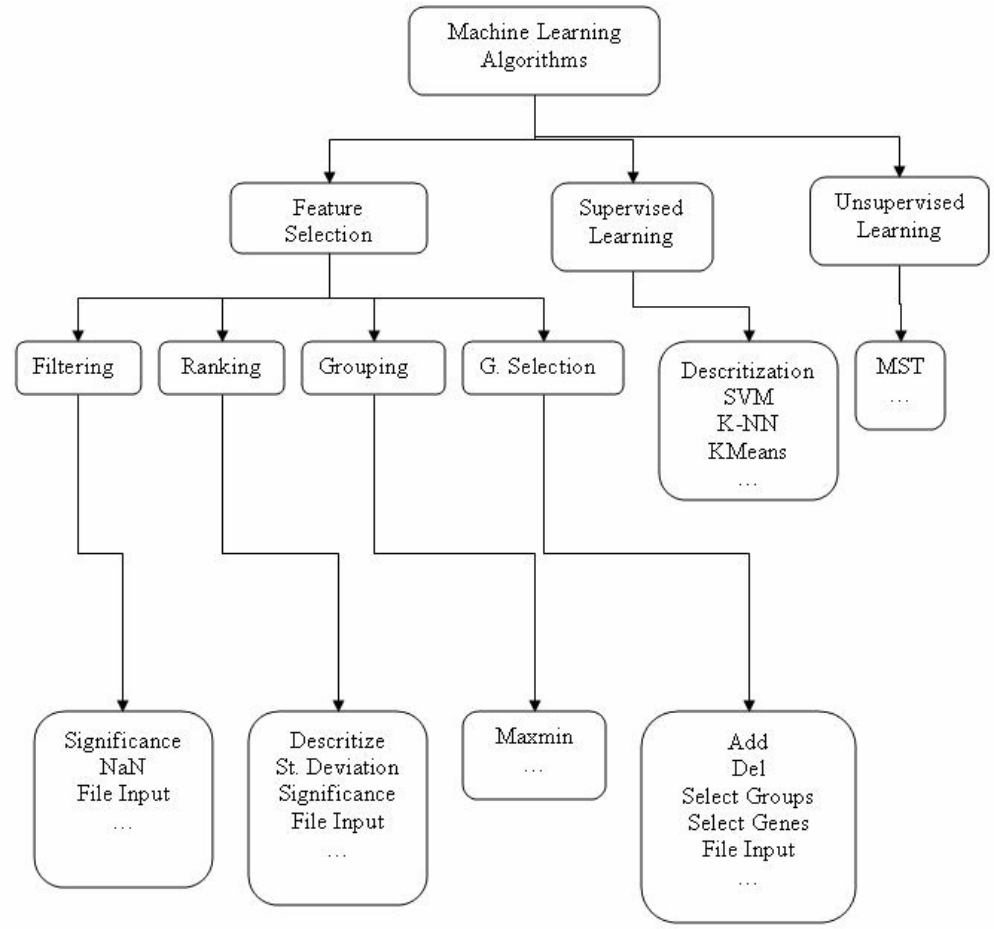


Figure 25. *Class Hierarchy of MineGene.*

6.1.3 General concerns of implementation

As we have seen, the general purpose machine learning tool should comprise with some certain requirements. One of them is that it should act as a plug-in in a gene expression database, thus it should be implemented in a general purpose, flexible computer language. Another concern is that it should be composed by several components with certain correlations between them. All the tasks presented before are families of certain algorithms (i.e., we have the family of gene ranking algorithms). Algorithms belonging to the same family share common attributes, methods and architecture. The only programming technique that ensures the component-like structure of the tool is the object oriented programming. Finally the

tool should utilize a Graphical User Interface (GUI) in order, for a user to have a visual contact with various possible algorithms and parameters of them.

The programming language that fulfills the above requirements is the C++ and the programming environment selected is Microsoft Visual Studio v. 6.0. The component based schema of the tool, depicted in figure 24, is reflected in the hierarchy of the classes as we can see in figure 25. It is crucial to note that MineGene's architecture allows a component / plug-in approach. Thus if a new specific (i.e., ranking) algorithm appears it is very easy and straight-forward to be embodied in the tool and enrich its architecture.

6.2 MineGene: A Guide to Operations

In this chapter we will describe the features and usage of MineGene. The initial GUI is presented in figure 26.

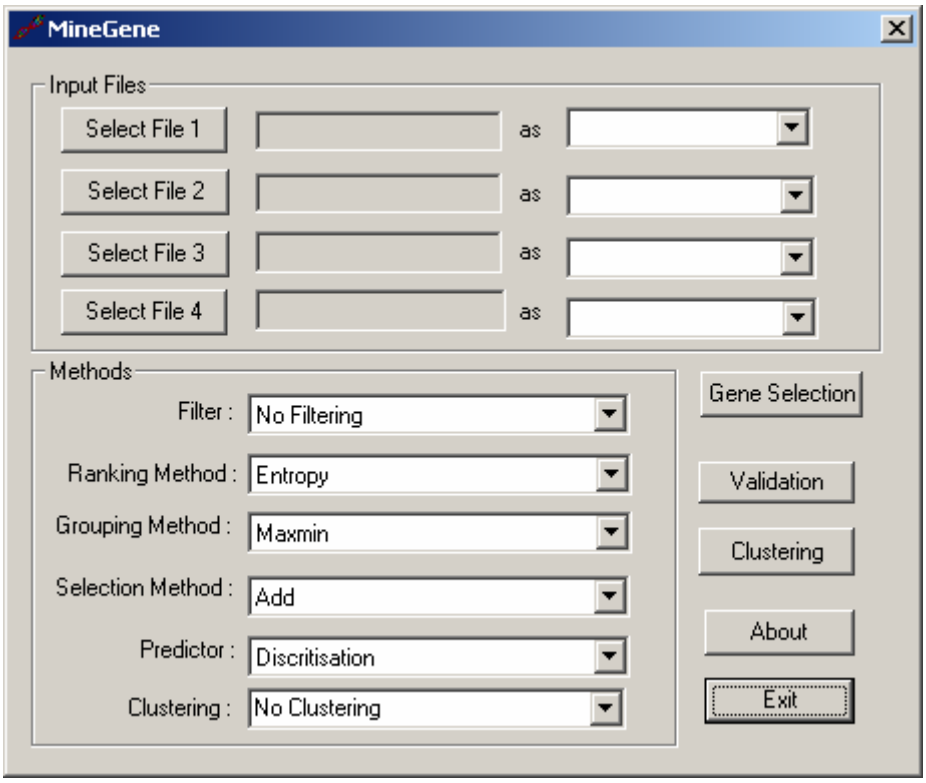


Figure 26. *MineGene's initial GUI.*

It is divided in three regions. The first named 'Input Files' where we select the input files to be processed. The second named 'Methods' contains all available methods organized as showed in chapter 6.1. Finally the third region contains buttons for manipulation.

6.2.1 Input Files

Each input file can be one of three types (see figure 27):

Train/Test File. These are files containing the primary data with gene expressions. They should be tab delimited files with *k* rows and *l* columns. In the i-th row and j-th column should be the expression of i-th gene of the j-th patient/sample. The filename could be anything, say "train.txt". The test file will be used only if we will not select LOOCV or Clustering elsewhere it will be ignored. The contents of the test files are used only to apply the learning method and assess its predictive ability.

Alongside with this file should be a file with the same name but with extension ".opt" ("train.opt" in our example). This file should contain the class assignment of each sample in the "train.txt" file as well as the name of each class. A typical ".opt" file could be:

```
classes     =     1 1 2 3 3 2 1 2 1 2 3 2 1 2 1 2 3 4 4  4 3 2 1 1
names       =     class1 class2 class3 class4
```

Another file that should exist is one with same name but with altered extension ".names" ("train.names" in our example). This file should contain the names of every genes plus arbitrary clustering information. It should have the following form:

```
0     AFFX-BioB-5_at      CL1
3     AFFX-BioB-M_at      CL2
1     AFFX-BioB-3_at      CL2
2     AFFX-BioC-5_at      CL1
```

The first column contains the consecutive number of a gene (starting from 0). The records in the file don't have to be sorted in any particular way, so with this consecutive number we hold the information of what gene is in each line. The second column contains the name of each gene. The third column contains pre-clustering information. We can assign a cluster value to each gene coming from an external source. This is useful when we perform our own clustering and we want to estimate our clustering efficiency according to an external cluster. Of course similarly we can estimate an external clustering.

If we select two or more files as train/test files then these files will be merged horizontally. The merged file will be used as a standalone train/test file. This is useful when we want to check the general content of a domain. Of course the two merged files should have the same number of lines.

Study file. When we select to build a classifier via the selected prediction method, the genes selected from the respective algorithm are printed in a separate file. It is very useful sometimes to compare the genes discovered by our algorithm with the genes discovered by an external study. So we provide the ability to select a file containing genes selected by a foreign study plus some kind of clustering information. This file can have any name and should have the following format:

```
0     AFFX-BioB-5_at      CL1
3     AFFX-BioB-M_at      CL2
1     AFFX-BioB-3_at      CL2
2     AFFX-BioC-5_at      CL1
```

The first column is ignored, although it is required for consistency reasons, as we wanted all the input files to have the same format. The second column contains the name of the genes of the external study and third column contains an arbitrary form of clustering. Whenever a study file is selected, at the end of the algorithm, the common genes are printed in a file with the same name as the train file but with altered extension ".common" (in our case the filename will be: "train.common"). This file has the following format:

```
3133   U38480_at     CL1    B
3297   U49114_at     CL2    B
6613   U68135_s_at   CL2    A
```

The first column contains the consecutive number of the common gene found (always starting from 0). The second column contains the name of the common gene. The third column contains clustering information contained in the ".names" file and the forth line contains clustering information contained in the study file.

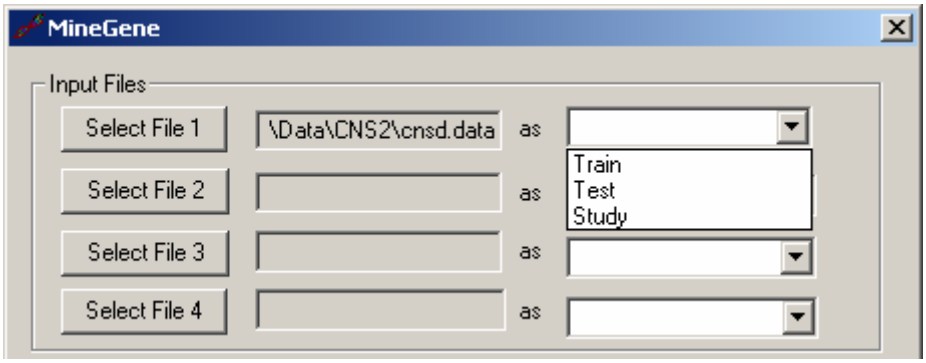


Figure 27. *Input Files.*

6.2.2 Methods

As it was described, we provide a series of families of algorithms, where each family is responsible for a certain aspect of the whole learning method. The provided methods, categorized in algorithmic families are:

Filtering. We provide the following filtering algorithms:

- *Remove null values* ('NaN'). As it was described in chapter 1.2.1.1 microarray expression data are sometimes erratic or non available. In these cases the gene expression matrix contains the value: NaN (Not a Number) in the corresponding position instead of a certain continuous value. Gene expression profiles containing too many NaN values do not exhibit any particular information, so they can be safely removed. With this algorithm we can remove gene expression profiles containing NaN values over a certain percentage.
- *Significance.* With this algorithm we can perform the consequent methods only to genes that are significantly regulated between the two sample classes. This was achieved using the Wilcoxon rank-sum test [162], which tests the null hypothesis that gene expression in one sample class is not different from

gene expression in the other tumor class. The user has to specify the maximum *p* value. Usually a value close to 0.05 yields satisfactory results.

- *Read from File.* We can select to perform our study restricted to genes whose names exist in an external file. The file should have the following format:

```
0    AFFX-BioB-5_at    CL1
3    AFFX-BioB-M_at    CL2
1    AFFX-BioB-3_at    CL2
```

  The first and third columns are ignored, although required for consistency reasons. The second column contains the names of the genes that will be furthermore processed. With this simple way we can restrict a study to be performed only in certain favorable genes.

- *No Filtering*. No filtering at all is performed. All genes are examined.

Ranking Method. We provide the following ranking methods.

- *Entropy*. The dicretization ranking method proposed in chapter 4.2.2.1.
- *Standard Deviation*. Gene discretization as proposed by Pomeroy et al [148]. This formula has been presented in the top of chapter 4.2.2.1.
- *Significance*. This ranking method is exactly as in the filtering family. Here the extracted probability is not used to decide if the gene will be neglected or not, but is assigned to it as ranking value.
- *Read from file*. Here we have the ability to assign a value to each gene that comes from an external source. The file should have the following format:

```
0    AFFX-BioB-5_at    13.5
3    AFFX-BioB-M_at    17.1
1    AFFX-BioB-3_at    21.2
```

  The first column is ignored. The second and third column contains the name of the file and its value respectively. If the file does not contain values for all available genes (all, except of these neglected from filtering) then an error message is appeared.

Grouping Method. After ranking genes are sorted according to their ranking in descending order. At the top of the ordering we have genes with maximum descriptive ability. We provide the following grouping method:

- *Maxmin*. Gene Grouping as presented in chapter 4.2.2.2.
- *No Grouping*. No grouping is performed at all. With this option, every gene is considered to belong to a unique group.

***Gene Selection*.** We provide the following gene selection methods:

- *Add*. The Add Gene Selection method as it was presented in chapter 4.2.2.2.
- *Del*. The Del Gene Selection method as it was presented in chapter 4.2.2.2.
- *Select Genes*. Apart from the heuristic/algorithmic methods for gene selection, a user can manually set the number of positive or negative ranked genes to be selected as markers. Let P and N denote the number of positive and negative ranked genes respectively. Via a special dialog box (figure 28) a user can set either the absolute number of desired positive and negative genes, as well as the percentage of them. We can also set to “lock” the ratio

of selected positive and negative genes to be P/N regardless our selection of genes. For example, let C be the number of absolute genes that we select to be our final markers. If we choose to "lock" the ratio then we will use $C\frac{P}{P+N}$ positive genes and $C\frac{N}{P+N}$ negative genes. Similarly if we select a percentage C instead of an absolute value, then we will use C% of P positive genes and C% of N negative genes. The same selection can be done to groups rather than genes.

- *Select Groups*. This method is exactly the same if the Select Genes method. Instead of manually selection of genes, we have the ability to select an arbitrary number of groups via the same dialog box and the same options.

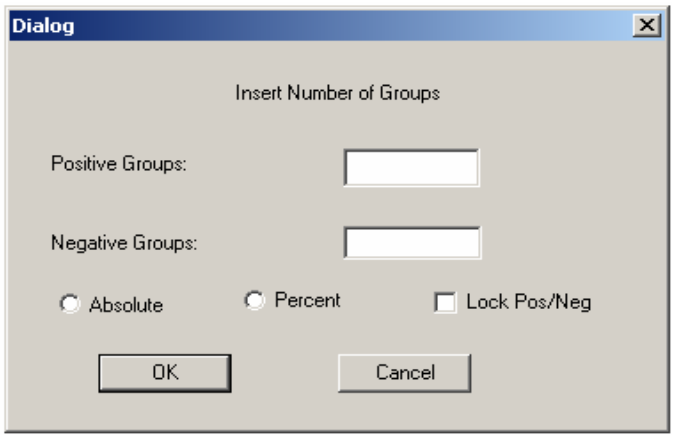


Figure 28. *Group Selection Dialog.*

- *Read from file*. Instead from applying a specific algorithm to find the most informative genes we can select genes from an external file. The file should have the following format:

```
0      AFFX-BioB-5_at      CL1
3      AFFX-BioB-M_at      CL2
1      AFFX-BioB-3_at      CL2
```

The first and third columns are ignored. The second column contains the names of the selected genes. This method is useful in order to estimate the descriptive ability of genes published in a foreign study.

Predictor. We provide the following prediction methods.

- *Discritisation.* The discritization prediction method as presented in 4.3.3.
- *SVM.* Performs the well-known SVM (Support Vector Machines [19]) prediction method. This method has a lot of parameters and options [205] that can be tuned from a special dialog box (figure 29).
- *KNN.* Performs the K-NN (K-Nearest Neighbors [164]). The number of nearest neighbors as well the distance method can be inserted via a special dialog box.
- *Kmeans.* Performs the KMEANS learning method.

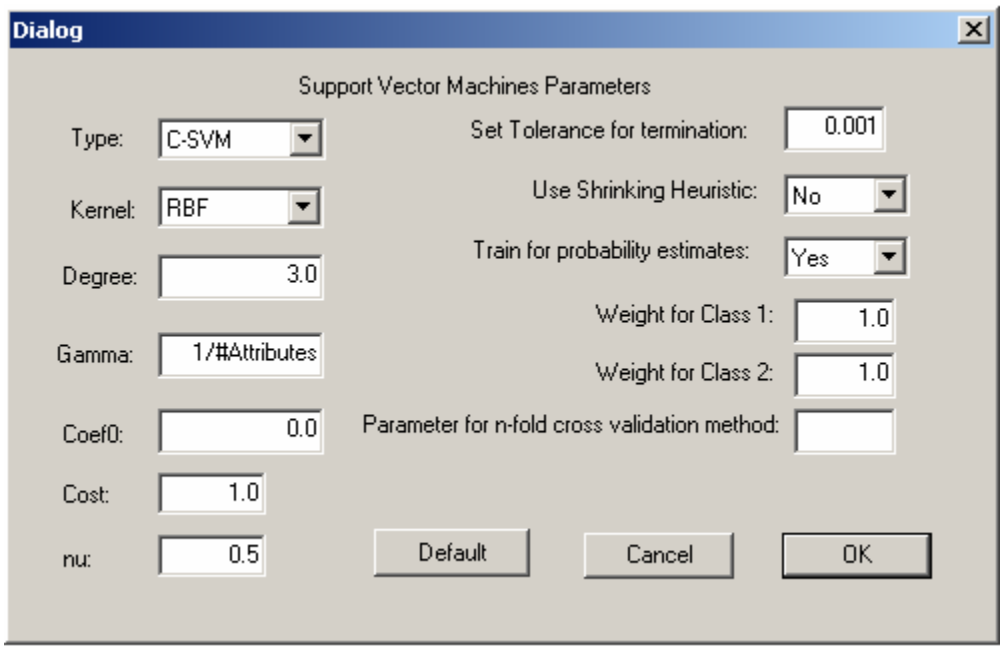


Figure 29. *Parameters of SVM package.*

Clustering. The only clustering method implemented is the Minimum Spanning Tree method proposed in chapter 5.2. The dialog box where we can set its options is in figure 30. In this dialog box we can set the MST method that will be followed (Prim's [193], Kruskal's [194] and Round Robin [195]). We can also set the distance method and the Category Utility as it was presented in chapter 5.2.2. We can also set some additional stopping criteria: The Minimum Cluster Members criterion ensures that each cluster should have a minimum number of objects. The percentage of Minimum Cluster Members criterion ensures that in each cluster a minimum percentage of total objects should participate. The maximum clusters criterion stops clustering when a certain number of clusters have been found and formed.

As presented in chapter 5.2.4 a special heuristic has been implemented in order to limit the number of stored distances. With Prune Distances percentage setting we can set the percentage of lower distances for each gene that should be computed. A reasonable value is 1%.

Additionally we can load an existing tree and apply clustering algorithm in this tree. It is not necessary to be a MST, any tree is suitable. To do this, we have to use the "Open Tree File" button. The file containing the tree must be formatted according the ".dot" format of Graphviz [206].

The "Open Dist File" button gives us the ability instead of calculating the distances to use an external file that contains all the distances of all the genes. This file should have the following format:

```
Gene1   Gene2  Dist1-2
Gene1   Gene3  Dist1-3
Gene2   Gene3  Dist2-3
```

The first and second column contains the names of the genes and the distance between them is in third column. This file must contain all possible distances between genes. With this option we can utilize the 'hybrid' characteristic of MST clustering algorithm, as described in chapter 5.2.

Finally, we can use an arbitrary graph instead of the fully connected graph, in order to produce the MST. This graph must be located in a file described according the ".dot" format of GraphViz and have to be chosen by the "Open Graph File" button.

The resulted clustering can be exported in JPEG format and visualized. Though, it is not recommended to visualize trees containing over than 1000 nodes as the inner JPEG exporting algorithm (GraphViz) has certain limitations.

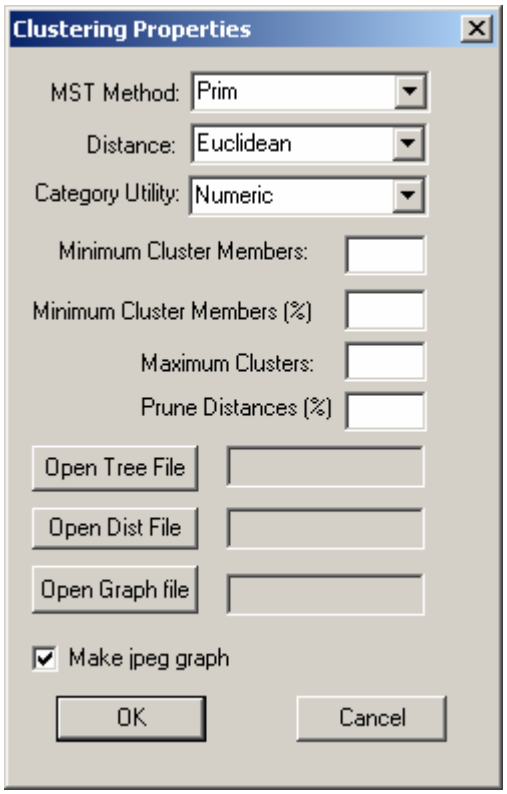


Figure 30. *MST Clustering Algorithm Properties.*

6.2.3 General usage

The button region of MineGene provides the ability to perform three fundamental data mining operation:

- **Gene Selection / Application of Learning Method.** This provides the ability to perform a supervised learning method that includes feature elimination / gene selection abilities. First a proper train and a test dataset have to be selected from the "Input Files" region. Then for each algorithm family (filtering, ranking, grouping, selection and predictor) one algorithm mast be chosen and set its parameters. The button "Gene Selection" has to be pressed.

- **LOOCV.** In order to perform LOOCV, we have to do make the same manipulations as in Gene Selection operation. We do not have to select a test file. The button “Validation” has to be pressed”.
- **Clustering.** In order to perform Clustering, we have to select an input Train File from the “Input Files” region. Then we have to set up a suitable filtering algorithm and a clustering algorithm. Then we have to press the “Clustering” button.

MineGene will inform when the algorithm terminates and will print the elapsed time. If the domain loaded has more than 2 classes, then the popup window in figure 31 pops-up.

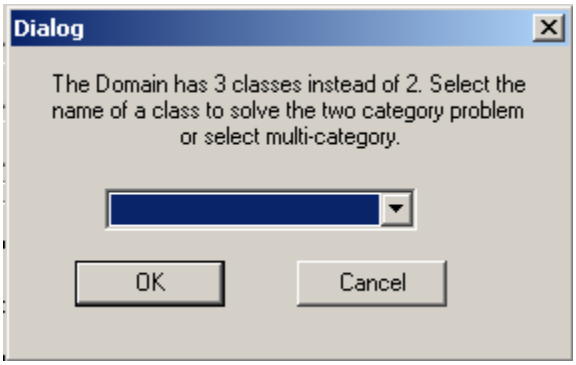


Figure 31. *Selecting between multi-class/category and two-category/class domains.*

From this popup window we can select a name of a class, for instance class1. Then all the samples not belonging to class1 will be grouped in the same class with name “Other” and the two category problem will be used. If we want to apply algorithms in a multi-category domain then we have to select the “Multi-category” option. Only the “Entropy”, “Select Groups”, “Select Genes” and “Discritization-Multi” methods work in multi-category domains.

6.2.4 Getting the Results: Output Files

At the end, when all specified algorithms have been completed the program produces a set of files containing various results and findings. All produced files are saved in a special directory named “Results”. The exported files are:

- `<train_filename>.selectedGenes`

  This file contains the genes selected from the selection algorithm. It looks like this:

```
3133      U38480_at     CL1
3297      U49114_at     CL2
6613      U68135_s_at   CL1
```

  The first column contains the consecutive number of the gene as it is appeared in the train/test file. It is important to note that the first gene in train/test file has consecutive number 0, instead of 1. The second column contains the name of the gene and the third columns contains its corresponding cluster information as it is contained in the “.names” file.

- `<train_filename>.classification`

  This file contains the classification results that produced by the prediction algorithm. It looks like this:

```
1
1
1
2
1
2
.
.
.
```

  On each row the assigned class is printed.

- `<train_filename>.results`

  It contains all the essential information and results of the whole process. Whether we have performed gene selection or clustering, this file has different form. If we have performed gene selection the file looks like this:

```
Train File = E:\Master\leukemia\train_tab.txt
Test  File = E:\Master\leukemia\test_tab.txt
Initial Genes = 7129
Genes after filtering = 7129
Train Samples = 38
Class: ALL has 27 train samples
Class: AML has 11 train samples
Test Samples = 35
Class: ALL, has 21 test samples
Class: AML, has 14 test samples
Number of Genes Finally Selected = 1
Number of Positive Ranked Selected Genes = 0
Number of Negative Ranked Selected Genes = 1
Number of Positive Groups Selected = 0
Number of Negative Groups Selected = 1
File with selected Genes = E:\Master\leukemia\train_tab.selectedGenes
File with Classification Results = E:\Master\leukemia\train_tab.classification
ALL / ALL   Classification = 19 54.285714%
ALL / AML   Classification = 2 5.714286%
AML / ALL   Classification = 1 2.857143%
AML / AML   Classification = 13 37.142857%
True  Classification = 32 91.428571%
False Classification = 3 8.571429%
```

  If we have performed clustering the file looks like this:

```
Clustering Results:
Train File = E:\Master\leukemia\leukemia3\train_tab2.txt
Initial Genes = 10
Genes after filtering = 10
Train Samples = 38
```

```
Clustering / Tags        Entropy: 0.857044
      Tags / Clustering  Entropy: 0.797797
Found 3 Clusters
Cluster 1 has 5 members
Cluster 2 has 3 members
Cluster 3 has 2 members
```

The Entropy values presented in this file are produced from the application of information gain formula in the clustering information yielded by the clustering algorithm and the clustering information contained in the ".names" file:

$$E = \sum_{i=1}^{\#CL} \frac{\#CL_i}{n} E(CL_i)$$

$$E(CL_i) = -\sum_{j=1}^{\#Cl} P(Cl_{ij}) \log P(Cl_{ij})$$

$$Cl_{ij} = \frac{\#Cl_{ij}}{\#CL_i}$$

Where *#CL* is the total number of clusters produced by our algorithm and *#Cl* is the total number of clusters contained in ".names" file. $\#CL_i$ is the number of genes contained in cluster *i* of our algorithm and $\#Cl_{ij}$ is the number of genes contained in cluster j of ".names" clustering and belong to cluster *i* of our algorithm.

- `<train_filename>.ranking`

  It contains the ranking of the genes produced by the ranking algorithm. It looks like this:

```
-297.000000  6613.000000  U68135_s_at
-286.000000  2849.000000  U17977_at
-275.000000  4542.000000  X74764_at
-275.000000  6165.000000  X83705_s_at
-275.000000  1747.000000  M16404_at
-270.000000  2628.000000  U04313_at
-270.000000  2696.000000  U09117_at
```

  The first column contains the ranking value of the gene. The second contains the consecutive number of the gene (always starting from 0). The third line contains the name of the gene. This file is sorted according the ranking values in ascending order. It is also printed the file <train_filename>.usranking that has the same information but is unsorted.

- `<train_filename>.bin` and `<test_filename>.bin`

  This file contains the expression values of all genes discretized according to the method proposed in chapter 4.2.2.1.

- `<train_filename>.binSelected` and `<test_filename>.binSelected`

  This file contains a discretization of the expression profile of the selected genes. The contents of this file are valuable, if we want to check the efficiency of the gene selection algorithm. A successful gene selection algorithm should select genes that are regulated significantly different among two class samples. The discretization should exhibit this regulation. The contents have the following format:

```
0    0    0    1    0    0    1    1    1    1
0    0    1    0    1    1    1    1    1    1
1    1    1    1    1    0    0    0    0    0
```

  These are the discretised values of the selected gene expression profiles. These values have been yielded from the following formula:

$$w = \frac{\max - \min}{n}$$

$$d_j = \begin{cases} n & , E_j = \max \\ \left[\frac{E_j - \max}{w} \right] + 1 & , else \end{cases}$$

  max and *min* are the minimum and maximum expression values of a selected gene. E_j is the expression value of a selected gene at the *j* sample and *n* is the total number of samples.

- `<train_filename>.genes`

  This file contains the genes that where processed, namely the genes that passed the filter. It has exactly the same format as the ".names" file.

- `<train_filename>.grouping`

  This file contains the grouping information of all genes generated by the respecting grouping algorithm. The file has the following format:

```
6613   U68135_s_at   1
2849   U17977_at     2
4542   X74764_at     3
```

  The first column contains the consecutive number of each gene. The second column contains the gene's name and the third column contains the corresponding group that the gene belongs.

- `<train_filename>.log`

  During runtime, several messages, events and progress status is printed in this file. This fire is useful in time consuming operations, especially during the clustering procedures. A timestamp is printed along which each message for example:

```
Fri Mar 11 15:41:39 2005 --> Starting MST procedure..
Fri Mar 11 15:41:51 2005 --> End of MST procedure
Fri Mar 11 15:41:51 2005 --> Start of clustering procedure
Fri Mar 11 15:41:51 2005 --> start of Finding primary best edge
Fri Mar 11 15:42:01 2005 --> Best Cut done: 10%  Total: 24188
Fri Mar 11 15:42:06 2005 --> Best Cut done: 20%  Total: 24188
…
```

6.3 Future Work for MineGene

The presented software toolkit, offered an integrated environment and a cohesive collection of machine learning algorithm for gene expression analysis through data mining.

- Although the major input can be easily acquired from gene expression databases through exported tab delimited files, we have to embed code supplied by MGED, and to follow specific directives in order to conform to MIAME guidelines. With this advance we will be able to feed with gene expression data our algorithms explicitly from gene expression databases, our results will be published in the same database system where the original data are laid and any researcher will be able to add his/her own algorithm to the existing toolkit schema.
- Although clinical applications and scenarios have already been presented we want to provide a full interconnection with a Clinical Information System (CLIS). Thus, a different clinical profile can be queried to the CLIS and produce distinct patients' ids. These ids can be used to export patient's expression signature from a gene expression database. From gene expression data mining we can extract specific differential gene regulations that designate the genotypic profile of patients. These differences can be studied further to gain intrinsic knowledge of the causing causes of the initial clinical differentiation and subsequently, to broaden medical research.
- As an addition, we could add existing or novel visualisation algorithms to gain insights of gene expressions regulations.

7. Conclusions and Future Work

7.1 Conclusions

We have presented the structural components of an integrated clinico-genomic environment where the genomic information mainly stemmed from microarray gene expression experiments is combined with information coming from clinical observations and processed via modern and novel machine learning algorithms. By analyzing gene expression profiles we expect to elaborate our knowledge about gene functional roles, genes inner-correlations and genes pathology. Outcomes from this utilization are expected to help healthcare specialists to infer critical deductions about the origins, pathology and treatment of several diseases affecting in various ways a vast part of population.

We surveyed microarray experiments, their usage and their essential role in gene expression profiling along with some certain difficulties that pose intrinsic challenges in machine learning researchers. Additionally, our approach was motivated by the construction of a seamless information system that acts as a microarray gene expression database incremented with machine learning application abilities. We surveyed existing genomic sequence and expression databases along with existing ontologies and annotations. The most cultivated and accepted ontology, MAGE, was analytical presented alongside with MIAME guidelines. Moreover we compared two of the most integrated and promising expression databases; ArrayExpress and BASE in various aspects to conclude that BASE is more suitable for our needs.

A vision of an integrated clinico-genomic environment where phenotypical profiles containing patient's clinical assessments are enriched with gene-expression profiles has been presented. Through data-mining algorithms that identify potential discriminatory genes we can indicate the molecular signature that best distinguish a specific phenotypic state. This signature - combined with clinical observations - can then be used for prognostic and therapeutic decision-making processes.

In the field of supervised learning methods, we presented an algorithm for gene ranking through an entropic metric according to their ability to distinguish between two sample classes. Genes then were sorted and grouped according to this ranking. A greedy gene selection / feature elimination methods was used to select the most discriminatory genes, with no use of any 'ad hoc' user presumption. Finally the same metric was used to build a predictor of unclassified samples. Our major contribution in this field was a gene selection via feature elimination algorithm based in groups addition and a sample class prediction method for multi-class domains. All these methods were applied to well-known datasets and their predictive accuracy were shown.

In the field of unsupervised learning methods we presented a novel Graph Theoretic Clustering algorithm. The distances coming either from the original expression values or from an external knowledge source are used to construct the fully connected graph of the genes. Then the Minimum Spanning Tree is extracted and an iterative hierarchical clustering algorithm is applied. The decision of whether to stop or continue clustering comes from a Category Utility that tests the compactness of potential clusters. The major deficiency of the algorithm; the space-demanding part of distance calculation was indicated and a heuristic that tackles this problem was proposed.

All the aforementioned methods plus some well-known methods for gene-ranking, filtering and predicting where implemented in a software tool named MineGene. MineGene is designed to serve as a machine-learning plug-in to a gene expression database. It has an extendable, components based architecture and it provides a usable GUI for maximum usability. Apart from the methods described above some additional methods where implemented. These methods include gene ranking and filtering methods based on significance estimation and a gene filtering method that eliminated the NaN (Not a Number) values. MineGine has the ability to import external data for gene filtering, gene ranking and gene selection and to compare the results with external studies. Finally creates a big variety of results ready to be used in other machine learning systems.

7.2 Future Work

Although we may be currently surprised by the advances of technology in genomic medicine, we can envisage the distant and not distant endeavours that have to be established in order to proceed.

In the field of genomic informatics we have to develop a comprehensive and comprehensible catalogue of all of the components encoded in the human genomes. So far we have specialized databases for expressions, sequences, proteins and pathways. We have to integrate these databases and to provide seamless information for every part of the human genome. Such elaboration will enable the prediction of protein function in the context of higher order processes such as the regulation of gene expression, metabolic pathways [207], [208] and signaling cascades. Moreover genomic databases have to be unified with clinical information systems, laboratory information systems and pathologo-anatomical information systems. The final objective of such higher-level functional analysis will be the elucidation of integrated mapping between genotype and phenotype [209]. These advances will narrow the difference between clinical and genomic domain with benefits in both sides.

As we moving toward the translation of genome-based knowledge into health benefits we have to define some common aims. First, we have to identify genes and pathways with role in health and disease, and determine how they interact with environmental factors. Secondly we have to develop, evaluate and apply genome-based diagnostic methods for the prediction of susceptibility to disease, the prediction of drug response, the early detection of illness and the accurate molecular classification of disease. Finally we have to deploy methods that catalyse the translation of genomic information into therapeutic advances.

In the machine learning / data mining field, we have to establish new approaches to solving problems, such as the identification of different features in a DNA sequence, the analysis of gene expression and regulation, the elucidation of protein structure and protein-protein interactions and the identification of the patterns of genetic variation in populations and the processes that produced those patterns. We also have to introduce methods to elucidate the effects of environmental (non-genetic) factors of gene-environment interactions on health and disease. Finally, although new improved database technologies facilitate the integration and visualisation of different genomic data types, little has been done in the construction of reusable machine learning software modules, easily exchangeable and tuned.

An additional data mining perspective is the exploitation of the knowledge stemmed from the enormous digital information cited in libraries, publications, conference proceedings, announcements and other sources of scientific material. Every algorithmic outcome and every result regarding genomic research have to be documented, supported and advocated with scientific publications. As information is constantly diffused in Internet we have to build novel machine learning methods to span scientific resources, perform term based comparisons and yield significant scientific support for our results.

Finally, it is crucial to define policy options, and their potential consequences, for the use of genomic information and for the ethical boundaries around genomic research [6]. It is indubitable the genetics and genomics can contribute understanding to many areas of biology, health and life. Although freedom of scientific inquiry has been cardinal feature of human progress, it is not unbounded. It is important for society to define the appropriate and inappropriate uses of genomics.

References

[1] The International Human Genome Sequencing Consortium. Initial sequencing of the human genome. Nature 409, 860-921 (2001).

[2] Alvis Brazma, Helen Parkinson, Thomas Schlitt, Mohammadreza Shojatalab. A quick introduction to elements of biology –cells, molecules, genes, functional genomics, microarrays Draft October 2001. EMBL.

[3] Lockhart, D.J and Winzeler,E.A. (2000) Genomics, gene expression and DNA arrays. Nature, 405, 827-836.

[4] Alan E. Guttmacher, M.D. 2001. The Human Genome Project: implications for health and public health. AMCHP Annual Meeting.

[5] Guttmacher, A.E. & Collins, F.S. Genomic medicine – A primer. N. Engl. J. Med. 347, 1512-1520 (2002).

[6] FS Collins, ED Green, AE Guttmacher, MS Guyer. 2003. A vision for the future of genomics research. US National Human Genome Research Institute. Nature, 2003

[7] Bowtell DDL. Options available – from start to finish – for obtaining expression data by microarray. Nature Genet 1999; 21: 25-32.

[8] Chen Y, Dougherty ER, Bittner ML. Ratio-based decisions and the quantitative analysis of cDNA microarrays images. J. Biomed Opt 1997;2: 364-374.

[9] Cheung VG, Morley M, Aguilar F, Massimi A, Kucherlapati R, Childs G. Making and reading microarrays. Nature Genet 1999; 21: 15-19.

[10] Ermolaeva O, Rastogi M, Pruitt KD, et al. Data management and analysis for gene expression arrays. Nature Genet 1998; 20: 19-23.

[11] Aach J, Rindone W, Church GM. Systematic management and analysis of yeast expression data. Genome Res 2000; 10: 431-445.

[12] Ringwald M, Eppig JT, Kadin JA, Richardson JE. GXD: a gene expression database for the laboratory mouse: current status and recent enhancements. Nucleic Acids Res 2000; 28: 115-119.

[13] Miller G, Fuchs R, Lai E. IMAGE cDNA clones, UniGene clustering, and ACeDB: an integrated resource for expressed sequence information. Genome Res 1997; 7: 1027-1032.

[14] D.E. Bassett, M.B. Eisen, M.S. Boguski, Gene expression informatics: it's all in your mine, Nature Genetics 21, Supplement 1 (1999) 51-55.

[15] Kanehisa M. Post-Genome Informatics. Oxford University Press: Oxford, 2000.

[16] Somogyi R, Sniegoski CA. Modeling the complexity of genetic networks: understanding multigenic and pleiotropic regulation. Complexity 1996; 1:45-63.

[17] Kanehisa M, Goto S. KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res 2000; 28: 29-34.

[18] Ashburner M, Ball CA, Blake JA, et al. Gene ontology: tool for the unification of biology. Nature Genet 2000; 25: 25-29.

[19] Cristianini, N., Shawe-Taylor, J. (2000). An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press.

[20] Duda R, Hart P, Stork D. (2000). Pattern Classification. John Wiley and Sons, New York.

[21] Schena, M. Shalon, D. Davis, R.W and Brown, P.O. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270(5235): 467-70.

[22] Human Genome Project. [online] Available: http://www.ornl.gov/hgmis/about.html, cited February 2002.

[23] The Dog Genome Project. [online] Available: http://mendel.berkeley.edu/dog.html

[24] Mouse Genome Sequencing Consortium [online] Available: http://www.ensembl.org/Mus_musculus/

[25] Quackenbush J: Computational analysis of microarray data. Nat Rev Genet 2001, 2:418-427.

[26] Lashkari, D. A., DeRisi, J. L., McCusker, J. H., Namath, A. F., Gentile, C., Hwang, S. Y.,

Brown, P. O. & Davis, R. W. (1997). Yeast microarrays for genome wide parallel genetic and gene expression analysis. Proc. Natl. Acad. Sci. USA, 94, 13057--13062.

[27] J.D. Watson, F.H Crick A Structure for Deoxyribose Nucleic Acid, Nature 1953, 171, 737.

[28] Dov Stekel, 2003, Microarray Bioinformatics, Cambridge University Press.

[29] Steen Knudsen. 2004, Guide to Analysis of DNA Microarray Data. Wiley , 2nd Edition.

[30] Affymetrix. [online]. http://www.affymetrix.com/index.affx

[31] Ben-Dor, A. and Yakhini, Z. (1999). Clustering gene expression patterns. Proceedings of the Third Annual International Conference on Computational Molecular Biology RECOMB-1999 pp.33-42. ACM Press, Lyon.

[32] Brazma, A., and Vilo, J. Gene Expression Data Analysis. FEBS Letters 480 (2000) 17-24.

[33] Yang YH, Buckley MJ, Speed TP. Analysis of cDNA microarray images. Brief Bioinform. 2001 Dec;2(4):341-9.

[34] Bioinformatics Glossary. [online] http://dna.uta.fi/xml/courses/glossary/glossary-items.xml

[35] Dutton, G. Gene Expression Data Mining. The Scientist, 16(20), 2002.

[36] Glonek, G.F., and Solomon, P.J. Factorial and time course designs for cDNA microarray experiments. Biostatistics, 5(1):89-111, 2004.

[37] Hwang, D., Schmitt, W.A., Stephanopoulos, G, and Stephanopoulos, G. Determination of minimum sample size and discriminatory expression patterns in microarray data. Bioinformatics, 18(9):1184-93, 2002.

[38] Kerr, M.K. Experimental design to make the most of microarray studies. Methods Mol Biol., 224:137-47, 2003.

[39] Kerr, M.K., Churchill, G.A. Experimental design for gene expression microarrays. Biostatistics, 2(2):183-201, 2001.

[40] Simon, R., Radmacher, M.D., and Dobbin, K. Design of studies using DNA microarrays. Genet Epidemiol. 23(1):21-36, 2002.

[41] Simon, R.M., and Dobbin, K. Experimental design of DNA microarray experiments. Biotechniques, Mar; Suppl: 16-21, 2003.

[42] Yang, Y.H., and Speed, T. Design issues for cDNA microarray experiments. Nat Rev Genet., 3(8):579-588, 2002.

[43] Leung, Y.F., Cavalieri, D. Fundamentals of cDNA microarray data analysis. Trends Genet. 19(11):649-59, 2003.

[44] Nadon, R., Shoemaker, J. Statistical issues with microarrays: processing and analysis. Trends Genet. 18(5):265-71, 2002.

[45] Schena, M., Heller, R.A., Theriault, T.P., Konrad, K., Lachenmeier, E. and Davis, R.W. Microarrays: biotechnology's discovery platform for functional genomics. Trends Biotechnol. 16(7):301-306, 1998.

[46] Sherlock, G. Analysis of large-scale gene expression data. Brief Bioinform. 2(4):350-62, 2001.

[47] Smyth GK, Yang YH, Speed T. Statistical issues in cDNA microarray data analysis. Methods Mol Biol., 224:111-36, 2003.

[48] Stratowa, C. and Wilgenbus, K.K. (1999). Gene expression profiling in drug discovery and development. Curr. Opin. Mol. Ther., 1(6):671-679, 1999.

[49] Zweiger, G. Knowledge discovery in gene-expression-microarray data: mining the information output of the genome. Trends Biotechnol. 17(11):429-436, 1999.

[50] Dudoit, S., Yang, Y.-H. Callow, M.C., and Speed, T.P. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Statistika Sinica, 12(1), 2002.

[51] Efron, B. Robbins, empirical Bayes, and microarrays. Dept. of Statistics, Stanford, Technical Report 2001-30B/219, 2001.

[52] Efron, B., Storey, J.D., and R. Tibshirani, R. Microarrays empirical Bayes methods, and false discovery rates. Dept. of Statistics, Stanford, Technical Report 2001-23B/217, 2001.

[53] Efron, B., Tibshirani, R., Storey, J.D., and V. Tusher, V. Empirical Bayes analysis of a microarray experiment. JASA, 96:1151-1160, 2001.

[54] Goss Tusher, V., Tibshirani, R., and G. Chu, G. Significance analysis of microarrays applied to the ionizing radiation response. PNAS, 98:5116-5121, 2001.

[55] Li, L., Weinberg, C.R., Darden, T.A., Pedersen, L.G.: Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics, 17(12):1131-1142, 2001.

[56] Lönnstedt, I., and Speed, T.P. Replicated microarray data. Statistika Sinica, 12(1), 2002.

[57] Newton, M.A., Kendziorski, C.M., Richmond, C.S., Blattner, F.R., and Tsui, K.W. On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. J. Comp. Biol., 8:37-52, 2001.

[58] Ting Lee, M.L., Kuo, F.C., Whitmore, G.A., and Sklar, J. Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. PNAS, 97:9834-9839, 2000.

[59] Ben-Dor, A., Bruhn, L., Nachman, I., Schummer, M., and Yakhini, Z., Tissue Classification with Gene Expression Profiles. Journal of Computational Biology, 7:601-620, 2000.

[60] Causton, H.C, Quackenbush, J., and Brazma, A. Microarray Gene Expression Data Analysis: A Beginner's Guide. Blackwell Publishing, 2003.

[61] Datta, Su., and Datta, So. Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics, 19(4):459-466, 2003.

[62] Cho, R.J. et al. A genome-wide transcriptional analysis of the mitotic cell cycle. Mol. Cell., 2:65-73, 1998.

[63] Dudoit, S. , Fridlyand, J. & Speed, T. P. Comparison of Discrimination Methods for the Classification of Tumors by Using Gene Expression Data (Dept. of Statistics, University of California, Berkeley, CA), Technical Report 576, 2000.

[64] Fraley, C., and Raftery, A.E. How many clusters? Which clustering method? Answers via model-based cluster analysis. The Computer Journal, 41:578-588, 1998.

[65] Herrero, J., Valencia, A., and Dopazo, J. A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics, 17:126-136, 2001.

[66] Kerr, M.K., and Churchill, G.A. Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. PNAS, 98:8961-8965, 2001.

[67] Lakhani, S.R., and Ashworth, A., Microarray and histopathological analysis of tumours: the future and the past? Nature Reviews Cancer, 1:151-157, 2001.

[68] Roth, F.P., Hughes, J.D., Estep, P.W. and Church, G.M. Finding DNA regulatory motifs within unaligned non-coding sequencesclustered by whole-genome mRNA quantitation. Nat. Biotechnol., 16:939–945, 1998.

[69] Speed, T. Statistical Analysis of Gene Expression Microarray Data. Chapman and Hall/CRC Press, 2003.

[70] Tibshirani, R., Hastie, T., Eisen, M., Ross, D., Botstein, D., Brown, P., Clustering methods for the analysis of DNA microarray data. 1999. [http://www-stat.stanford.edu/~tibs/ftp/sjcgs.ps]

[71] Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. Diagnosis of multiple cancer types by shrunken centroids of gene expression. PNAS, 99(10):6567-6572, 2002.

[72] Yeung, K.Y., Fraley, C., Murua, A., Raftery, A.E., and Ruzzo, W.L. Model-based clustering and data transformations for gene expression data. Bioinformatics, 17:977-987, 2001.

[73] EMBL Nucleotide Sequence Database. [online] http://www.ebi.ac.uk/embl/

[74] Kanz,C., Aldebert,P., Althorpe,N., Baker,W., Baldwin,A., Bates,K., Browne,P., van den Broek,A., Castro,M., Cochrane,G. et al. ((2005)) The EMBL Nucleotide Sequence Database. Nucleic Acids Res., , 33, , D29–D33.

[75] GenBank Overview [online] http://www.ncbi.nlm.nih.gov/Genbank/

[76] National Institutes of Health (NIH) [online] http://www.nih.gov/

[77] Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Rapp, B.A. and Wheeler, D.L. (2000). Genbank. Nucleic Acids Res. 28, 1, 15-18

[78] National Center for Biotechnology Information (2000). Genbank Overview. [online] http://www.ncbi.nlm.nih.gov/Genbank/GenbankOverview.html
[79] DDBJ, DNA Data Bank of Japan. [online] http://www.ddbj.nig.ac.jp/
[80] UniGene Overview. [online] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene
[81] Pontius JU, Wagner L, Schuler GD. UniGene: a unified view of the transcriptome. In: The NCBI Handbook. Bethesda (MD): National Center for Biotechnology Information; 2003.
[82] TIGR Gene Indices. The Institute of Genomic Research. [online] http://www.tigr.org/tdb/tgi/
[83] Pertea, G., Huang, X., Liang, F., Antonescu, V., Sultana, R., Karamycheva, S., Lee, Y., White, J., Cheung, F., Parvizi, B., Tsai, J. and Quackenbush, J. (2003). TIGR Gene Indices clustering tools (TGICL): a software system for fast clustering of large EST datasets. Bioinformatics 19(5): 651-2.
[84] The I.M.A.G.E Cosortium. [online] http://image.llnl.gov/
[85] NCBI Reference Sequence (RefSeq). [online] http://www.ncbi.nlm.nih.gov/RefSeq/
[86] NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins Pruitt KD, Tatusova, T, Maglott DR Nucleic Acids Res 2005 Jan 1;33(1):D501-D504
[87] Ensembl Genome Browser. [online] http://www.ensembl.org/
[88] Ewan Birney et al. An Overview of Ensembl Genome Res. 2004 14: 925-928.
[89] TIGR-CMR. [online] http://www.tigr.org/tigr-scripts/CMR2/CMRHomePage.spl
[90] John L. Houle et. al., White Paper: Database Mining in the Human Genome Initiative, AMITA Corp. 2000
[91] Carulli, J.P. Artinger, M., Swain, P.M., Root, C.D., Chee, L., Tulig, C., Guerin, J., Osborne, M., Stein, G., Lian, J. and Lomedico, P.T. (1998). High throughput analysis of differential gene expression. J. Cell Biochem. 30-31, Supplement 0, 286-296
[92] MAGE-ML [online] http://www.mged.org/Workgroups/MAGE/mage-ml.html
[93] MGED 1st meeting [online] http://www.mged.org/Meetings/m1/index.html
[94] MGED Network. Ontology Working Group [online] http://mged.sourceforge.net/ontologies/index.php
[95] MIAME [online] http://www.mged.org/Workgroups/MIAME/miame.html
[96] MAGE. [online] http://www.mged.org/Workgroups/MAGE/mage.html
[97] Object Management Group. [online] http://www.omg.com/
[98] MAGE-OM. [online] http://www.mged.org/Workgroups/MAGE/mage-om.html
[99] MAGE-ML [online] http://www.mged.org/Workgroups/MAGE/mage-ml.html
[100] UML. [online] http://www.uml.org/
[101] eXtensible Markup Language (XML). [online] http://www.w3.org/XML/
[102] World Wide Web Consortium. [online]. http://www.w3.org/
[103] W3C Document Object Model. [online]. http://www.w3.org/DOM
[104] SAX. [online]. http://www.saxproject.org/
[105] Xquery 1.0: An XML Query Language. [online]. http://www.w3.org/TR/xquery/
[106] XML Path Language (XPath). [online] http://www.w3.org/TR/xpath
[107] Serial Analysis of Gene Expression. [online] http://www.sagenet.org
[108] Velculescu, V. E., Zhang, L., Vogelstein, B., and Kinzler, K. W. (1995). Serial Analysis Of Gene Expression. Science 270, 484-487.
[109] ArrayExpress [online] Available: http://www.ebi.ac.uk/arrayexpress/
[110] NCGR. [online] http://www.ncgr.org/
[111] Rosetta Inpharmatics - genomic research and data analysis. [online] http://www.rii.com/
[112] Agilent Technologies. [online]. http://www.agilent.com
[113] Shoemaker D.D et al. “Experimental annotation of the human genome using microarray technology” Nature 409, 922 - 927 (15 February 2001)
[114] The Tree of Life project. [online] http://phylogeny.arizona.edu/tree/life.html
[115] Species 2000. [online] http://www.sp2000.org/
[116] International Organization for Plant Information [online] http://iopi.csu.edu.au/iopi/

[117] Integrated Taxonomic Information System. [online]. http://iopi.csu.edu.au/iopi/
[118] NCBI Taxonomy [online] http://www.ncbi.nlm.nih.gov/Taxonomy/
[119] UniProt/Swiss-Prot [online] http://www.ebi.ac.uk/swissprot/
[120] UniProt/TrEMBL [online] http://www.ebi.ac.uk/trembl/
[121] Gene Ontology [online] http://www.geneontology.org/
[122] MGED Network :: MGED Ontology. [online]. http://mged.sourceforge.net/ontologies/MGEDontology.php
[123] About IMBB. [online] http://www.imbb.forth.gr/
[124] Forth – Ics. [online] http://www.ics.forth.gr/
[125] European Bioinformatics Institute [online] Available: http://www.ebi.ac.uk/
[126] Edgar R., Domrachev, M. and Lash, A.E. (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res., 30, 207-210
[127] Ball C.A., Brazma,A., Causton,H., Chervitz,S., Edgar,R., Hingamp,P., Matese,J.C., Parkinson,H.E., Quackenbush,J., Ringwald,M., Sansone,S., Sherlock,G., Spellman,P., Stoeckert,C., Tateno,Y., Taylor,R., White,J. and Winegarden,N. (2004) Submission of microarray data to public repositories. PLoS Biol., 2, 1276–1277.
[128] EBI Submissions – MIAMExpress. [online] http://www.ebi.ac.uk/miamexpress/
[129] Expression Profiles [online] http://ep.ebi.ac.uk/EP/
[130] Parkinson H. et al. ArrayExpress—a public repository for microarray gene expression data at the EBI. Nucleic Acids Res. 2005 January 1; 33(Database Issue): D553–D555.
[131] Lao H. Saal, Carl Troein, Johan Vallon-Christersson, Sofia Gruvberger, Åke Borg and Carsten Peterson BioArray Software Environment: A Platform for Comprehensive Management and Analysis of Microarray Data Genome Biology 2002 3(8): software0003.1-0003.6
[132] BASE Project Site. [online] http://base.thep.lu.se/
[133] GNU General Public License - GNU Project - Free Software Foundation (FSF). [online] http://www.gnu.org/licenses/gpl.html
[134] PHP: Hypertext preprocessor [online] http://www.php.net
[135] MySQL [online] http://www.mysql.com
[136] Apache HTTP Server Project [online] http://httpd.apache.org
[137] J. Celis, Proteomics and Functional Genomics in Translational Cancer Research: toward an integrated approach. Presentation in Cancer: Molecular Targets for novel Therapies, 3rd Simposio Scientifico, Pabellón San Carlos, Hospital Clínico, Madrid, April 25 & 26, 2003.
[138] V. Maojo, I. Iakovidis, F. Martin-Sánchez, J. Crespo, C. Kulikoswki, Medical Informatics and Bioinformatics: European efforts to facilitate synergy, Journal of Biomedical Informatics 34:6 (2001) 423-427.
[139] G. Potamias, Utilizing Gene Functional Classification in Microarray Data Analysis: a Hybrid Clustering Approach, 9th Panhellenic Conference in Informatics, 21-23 November, Thessaloniki, Greece, 2003.
[140] G.S Ginsburg, J.J. McCarthy, Personalized medicine: revolutionizing drug discovery and patient care, Trends Biotechnol 19 (12) (2001) 491-496.
[141] H.F. Friend, How DNA microarrays and expression profiling will affect clinical practice, Br Med J. 319 (1999) 1-2.
[142] Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. and Lander, E.S. (1999) Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science 286, 531-537. http://www-genome.wi.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43
[143] van't Veer, L., Dai, H., Vijver, M.v.D., He, Y., Hart, A., Moa, M., Peterse, H., Kooy, K.v.D., Marton, M., Witteveen, A., Schreiber, G., Kerkhoven, R., Roberts, C., Linsley, P., Bernards, R., and Friend, S. (2002). Gene expression profiling predicts clinical outcome of breast cancer. Nature 415, 530–536. http://www.rii.com/publications/2002/vantveer.htm

[144] Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D. and Levine, A.J. Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. (1999) Proc. Natl. Acad. Sci. USA 96, 6745-6750. http://microarray.princeton.edu/oncology/affydata/index.html

[145] K. Fuzarewicz, M. Weinch. Selecting differentially expressed genes for colon classification, Int. J. Appl. Math. Comput. Sci. 13 (3) (2003) 327-335.

[146] Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511. http://llmpp.nih.gov/lymphoma/

[147] Hedenfalk et al, Gene-expression profiles in hereditary breast cancer, N Engl J Med 344 (2001) 539-548.http://research.nhgri.nih.gov/microarray/NEJM_Supplement/

[148] Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., Sturla, L.M., Angelo, M., McLaughlin, M.E., Kim, J.Y., Goumnerova, L.C., Black, P.M., Lau, C., et al. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature 415, 436–442. http://www-genome.wi.mit.edu/mp3/CNS

[149] Troyanskaya, M.E. Garber, P.O. Brown, D. Botstein, R.B. Altman, Nonparametric methods for identifying differentially expressed genes in microarray data. Bioinformatics 18 (11) (2002) 1454-1461.

[150] The future of cancer treatment. [online] http://www.economist.com/science/displayStory.cfm?story_id=3285968

[151] Definition of Biomedical Informatics. [online]. http://faculty.washington.edu/gennari/MedicalInformaticsDef.html

[152] Reich, D.E. & Lander, E.S. On the allelic spectrum of human disease. Trends Genet. 17, 502-510 (2001).

[153] E.J. Yeoh et al, Classification, subtype, and prediction of outcome in pediatric acute leukaemia by gene expression profiling, Cancer Cell 1 (2) (2002) 133-143.

[154] Sorlie, T., Perou, C., Tibshiranie, R., Aasf, T., Geislerg, S., Johnsen, H., Hastiee, T., Eisen, M., Van de Rijni, M., Jeffreyj, S., Thorsenk, T., Quistl, H., Matesec, J., Brown, P., Botstein, D., Lonningg, P.E., and Borresen-Dale, A., Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications, Proc. Natl. Acad. Sci. US, 98 (19) (2001) 10969-10874.

[155] Bioinfomed. [online]. http://bioinfomed.isciii.es/

[156] John L. Houle et. al., White Paper: Database Mining in the Human Genome Initiative, AMITA Corp. 2000.

[157] Wu TD. Analysing gene expression data from DNA microarrays to identify candidate genes. J Pathol95 :53 –65,2001 .

[158] Brazma, A. and Vilo, J. 2000. Gene expression data analysis. FEBS Lett. 480: 17–24.

[159] Brown. M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares, M.J. and Haussler, D. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines .Proc. Natl. Acad. Sci. USA 97, 262-267.

[160] Eisen, M., Spellman, P.T., Botstein, D. and Brown, P.O. (1998) Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA 96, 14863-14867.

[161] Su, Andrew I., Welsh, John B., Sapinoso, Lisa M., Kern, Suzanne G., Dimitrov, Petre, Lapp, Hilmar, Schultz, Peter G., Powell, Steven M., Moskaluk, Christopher A., Frierson, Henry F., Jr., Hampton, Garret M. Molecular Classification of Human Carcinomas by Use of Gene Expression Signatures Cancer Res 2001 61: 7388-7393

[162] Warren Chase, Bown Fred. “General Statistics”, (1996). John Wiley & Sons, New York.

[163] Stone, M. Cross-validation choice and assessment of statistical predictions. J.R. Stat. Soc. B-36:111-114, 1974.

[164] Dasarathy, V.B. (ed). Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques (IEEE Computer Society Press, Los Alamitos, California, 1991)

[165] Potamias, G., Koumakis, L., Moustakis, L. Gene Selection via Discretized Gene-Expression Profiles and Greedy Feature-Elimination. 2004. Hellenic Conference on Artificial Intelligence.

[166] Brazma, A., Jonassen, I., Vilo, J. and Ukkonen, E. (1998) Predicting Gene Regulatory Elements in Silico on a Genomic Scale. Genome Res. 8, 1202-1215.

[167] Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, S. Levy, A Comprehensive Evaluation of Multicategory Classification Methods for Microarray Gene Expression Cancer Diagnosis, Bioinformatics (Advance Access) published September 16, 2004.

[168] U. Fayyad, K. Irani, Multi-interval discretisation of continuous-valued attributes for classification learning, 13th International Joint Conference of Artificial Intelligence, Morgan Kaufmann, San Francisco, CA 1022-1029, 2003.

[169] J. Li, L. Wong, Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns, Bioinformatics 18(2002) 725-734.

[170] M.A. Hall, Correlation-based Feature Selection for Machine Learning, PhD thesis, University of Waikato (1999).

[171] R. Kohavi, G.John, G, Wrappers for feature subset selection, Artificial Intelligence (special issue on Relevance) 97 (1-2) (1996) 273-324.

[172] P.W. Baim, A Method for Attribute Selection in Inductive Learning Systems, IEEE PAMI 10 (6) (1988) 888-896.

[173] L.G. Valiant, A theory of the learnable, CACM, 27 (11) (1984) 1134-1142.

[174] D. Angluin. Queries and concept learning, Machine Learning 2 (1998) 319-342.

[175] Blum. On-line algorithms in machine learning. [online] http://www.cs.cmu.edu/~avrim/Papers/pubs.html .

[176] R. Rivest. Learning decision lists, Machine Learning 2 (3) (1987) 229-246.

[177] Dhagat, L. Hellerstein. Pac learning with irrelevant attributes, Proceedings of the Thirty-Fifth Annual Symposium on Foundations of Computer Science, 64-74, 1994.

[178] Hartigan, J.A. (1975) Clustering Algorithms, John Wiley and Sons, New York

[179] Tavazoie, S., Hughes, D., Campbell, M.J. Cho, R.J. and Church, G.M. (1999) Systematic determination of genetic network architecture. Nature Genet 22, 281-285.

[180] Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. and Golub, T. (1999) Interpreting patterns of gene expression with self-organizing maps. Proc. Natl. Acad. Sci. USA 96, 2907-2912

[181] DeRisi, J.L., Iyer, V.R. and Brown, P.O. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. Science 278, 680-686.

[182] van Helden, J., Andre, B. and Collado-Vides, J. (1998). Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. J. Mol. Biol. 281, 827-842.

[183] Chu, S., DeRisi, J.L., Eisen, M., Mulholland, J., Botstein, D., Brown, P.O. and Herskowitz, I. (1998) The transcriptional program of sporulation in budding yeast. Science 282, 699-705.

[184] Spellman, P.T., Sherlock, G., Zhangm M., Iyer, V.R., Amders, K., Eisen, M., Brown, P.O., Botstein, D. and Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol. Biol. Cell 9, 3273.

[185] Holstege, F., Jennings, E., Wyrick, J., Lee, T., Hengartner, C., Green, M., Golub, T., Lander, E. and Young, R. (1998). Dissecting the Regulatory Circuitry of a Eukaryotic Genome. Cell 95, 717-728.

[186] Iyer, V.R., Eisen, M.B., Ross, D.T., Dchuler, G., Moore, T., Lee, J.Cf., Trent, J.M., Staudt, L.M., Hudson Jr., Boguski, M.S., Lashkari, D., Shalon, D., Botstein, D. and Brown, P.O. (1999) The transcriptional program in the response of human fibroblasts to serum. Science 283, 83-87.

[187] Lee, C., Klopp, R.G., Weindruch, R. and Prolla, T.A. (1999). Gene expression profile of aging and its retardation by caloric restriction. Science 285, 1390-1393.

[188] Ben-Dor, A., Bruhn, L., Friedmanm N., Nachman, I., Schummer, M. and Yakhini, Z. (2000) The Fourth Annual International Conference on Computational Molecular Biology RECOMB-2000, pp. 54-64, ACM Press, Tokyo.

[189] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comp. 20 _1971) 68-86.

[190] R.L. Page, Minimal spanning tree clustering method (ACM algorithm 479), CACM 17 (6)

(1974) 321-323
[191] D. Xu, V. Olman, L. Wang, Y. Xu, EXCAVATOR: A computer program for efficiently mining gene expression data, Nucleic Acids Res. 31 (19) (2003) 5582-5589.
[192] Introduction to EXCAVATOR. [online]. http://compbio.ornl.gov/structure/excavator/
[193] R. Prim, Shortest connection networks and some generalizations, Bell Syst. Tech. J. 36 (1957) 1389-1401.
[194] Kruskal, J. B. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proc. Amer. Math. Soc. 7, 48-50, 1956.
[195] Gabow, H. N. A framework for cost-scaling algorithms for submodular flow problems. In Proceedings of the 34th Annual Symposium on the Foundations of Computer Science (1993). IEEE, IEEE Computer Society Press, pp. 449-458
[196] Gennari, J., Langley, P., and Fisher, D. (1989). Models of incremental concept formation. Artificial Intelligence, 40, (pp.11-62).
[197] J. Yoo and S. Yoo."Concept Formation in Numeric Domains. Proceedings of Computer Science Conference, pp. 36-41, Nashville, TN, March, 1995.
[198] D. Epstein, Fast hierarchical clustering and other applications of dynamic closets pairs, J. ACM Exp. Algorithms 5 (2000) 1-23.
[199] Wen, X., Fuhrman, S., Michaels, G. S., Carr, D. B., Smith, S., Barker, J. L., & Somogyi, R. Large-scale temporal gene expression mapping of central nervous system development. (1998). Proc. Natl. Acad. Sci. USA 95, 334-339.
[200] D. Watson, T.Martinez, Improved heterogeneous distance functions, J. Artif. Intell. Res. 6 (1997) 1-34.
[201] Potamias, G. (2002). Distance and Feature-Based Clustering of Time Series: An Application on Neurophysiology. SETN, LNAI 2308, pp. 237-248, 2002.
[202] Potamias, G. Dermon R.C. (2004). Protein synthesis profiling in the developing brain: a graph theoretic clustering approach. Computer Methods and Programs in Biomedicine. Article in press.
[203] Silicon Genetics Products: GeneSpring. [online] http://www.silicongenetics.com/cgi/SiG.cgi/Products/GeneSpring/index.smf
[204] MolMine. [online] http://www.molmine.com/frameset/frm_jexpress2.asp
[205] LIBSVM. [online] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[206] Graphviz [online] http://www.graphviz.org/
[207] Förster, J., Gombert, A.K., and Nielsen, J. 2002. A functional genomics approach using metabolomics and in silico pathway analysis. Biotechnol. Bioeng. 79: 703-712.
[208] King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G. K., Bryant, C. H., Muggleton, S. H., Kell, D. B., Oliver, S. G.: Functional genomic hypothesis generation and experimentation by a robot scientist. In: Nature 427: 247–252 (2004)
[209] Bork, P., Dandekar, T., Diaz-Lazcoz, Y., Eisenhaber, F., Huynen, M. and Yan, Y. (1998). Predicting function: from genes to genomes and back. J. Mol. Biol. 283, 4, 707-725.
[210] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (81) (1986) 81-106.

Appendix A. MIAME Guidelines Description

A.1 MIAME, array design description.

1. *Array*-related information:
    - Array design name.
    - Platform type. Whether the array is in-situ synthesized, spotted or some other type of array.
    - Surface and coating specification. The physical composition of the array (nylon or glass) and description of any chemical derivitisation on the surface of the array.
    - Physical dimensions of the array.
    - Number of features on the array. Includes the number of features in both x and y, and details of any grids on the array.
    - Availability. Name of supplier and catalogue number for commercial arrays, or production protocol for custom-made arrays.
2. *Reporter* type information:
    - Type of reporter. Whether the reporters are synthetic oligonucleotides, PCR products, plasmids, colonies or other.
    - Single- or double- stranded.
    - For each reporter:
        - Sequence or PCR information. The sequence if known (e.g. oligonucleotides), sequence accession number or primer pairs (if relevant).
        - Exact or approximate length of sequence.
        - Clone information. If relevant, the clone ID, clone provider, date of provision and availability of the clone.
        - Element generation protocol. Sufficient information to reproduce the element on custom arrays that are not generally available.
3. *Feature* type information:
    - Dimensions. The physical size of the features.
    - Attachment. Covalent, ionic or other. If the feature is an oligonicleotide, whether attachment is from 3' or 5' end of oligonucleotide.
    - For each feature:
        - Location on the array. Both physical and logical coordinates.
        - Which reporters. Which reporter sequence is on the feature,
4. For each *composite* sequence:
    - Which reporters it contains.
    - The reference sequence.
    - Gene or EST names. Including links to appropriate databases (e.g., UniGene or RefSeq).
5. *Control* elements on the array:
    - Position of the feature. Logical coordinates.
    - Control type. Spiking, normalization, negative or positive.
    - Control qualifier. Endogenous or exogenous.

A.2 MIAME - Experimental Description

The aim of the experimental description is to give sufficient information that another laboratory would be able to repeat the experiment. An experiment may consist of one or more hybridizations to one or more types of array. The experimental description is broken into four main parts, each of which has several components:

I. *Experimental design*.
- Authors, laboratory and contact information.
- Type of experiment. Typical experiments might be normal vs. disease comparison, treated vs. untreated comparison, time course or dose response.
- Experimental factors. These are the parameters or conditions that are tested in experiment. For example, treatment, time, dose or genetic variation.
- Number of hybridizations in the experiment.
- Whether or not a common reference sample has been used.
- Quality control steps. These include replications at different levels, the use of dye reversal, or the inclusion of quality control features.
- Description of experiment and its goal.
- Links to journal and/or web publication of the experiment.
- Journal or URL citations.

II. *Samples used, extract preparation and labelling*.

MIAME devised a hierarchical terminology for describing the samples that are hybridized to arrays.
- *Biosource properties*. The biosource is the term used to describe to organism from m which the sample that will be hybridized to the array is derived. It has the following properties:
  - Organism. Names are used from the NCBI taxonomy.
  - Contact details. Who to contact for information about the sample.
  - Descriptors relevant to the sample.
    - Sex, e.g. male, female, hermaphrodite.
    - Age. Including relevant units (days, months, years) and whatever from birth or embryolysis.
    - Developmental stage. An organism could develop at different rated depending on environmental conditions so this is included in addition to age.
    - Organism part. Tissue.
    - Cell type.
    - Animal/plant strain or line.
    - Genetic variation, e.g., wild-type, gene knockout or transgenic variation.
    - Individual genetic characteristics. Disease associated alleles or polymorphisms.
    - Additional clinical information.
    - Individual ID.

- *Biomaterial manipulations*. These are the laboratory processes carried out to the biosource as part of the experiment. They include:
    - Growth conditions.
    - In vivo treatments.
    - In vitro treatments, including cell culture conditions.
    - Treatment type, e.g. small molecule (drug), heat shock, food deprivation.
    - Separation technique, e.g. none, microdissection, FACS.
- *Hybridiation extract preparation protocol*. This is the nucleic acid that is extracted from the biomaterial that will be labeled:
    - Extraction method, e.g., URL of protocol.
    - Extract type, e.g. total RNA, mRNA or genomic DNA.
    - Amplification, e.g., RNA polymerases or PCR.
- *Labeling protocol*. For each extract:
    - Amount of nucleic acid labeled.
    - Label used, e.g., A-Cy3, G-Cy5 or 33P.
    - Label incorporation method, e.g., URL of protocol.
- *External controls added to hybridization extract*. These are spiking controls added for quality control purposes.
    - Element on array expected to hybridize to spiking control.
    - Spike type, e.g., oligonucleotide or bacterial DNA.
    - Spike qualifier, e.g., concentration, expected ration or labeling methods.

III. *Hybridization procedures and parameters*.

- Information about which *labelled extracts* have been *hybridized* to which arrays. The labelled extracts relate to the sample, and the array will relate to array design information.
- *Hybridization protocol*. This would normally include
    - The solution, e.g., Na^+ concentration or formamide concentration.
    - Blocking agent, e.g., COT1.
    - Wash procedure, e.g., temperature and Na^+ concentration.
    - Quantity of labeled target used.
    - Time, concentration, volume and temperature.
    - Hybridization instruments, e.g., manufacturer and model.

IV. Measurement data and specifications of data processing.

MIAME provides standards for describing the data from a microarray experiment at three levels. At the lowest level, the raw data is the image of the array. The second level is the image quantitation table, which contains the information produced by the feature extraction software such as mean intensity, number if pixels and pixel standard deviation. At the highest level, gene expression measurements from all the arrays in the experiment are normalized and combined to produce a gene expression measurement table for the experiment.

- *Raw data description*. The protocols and settings for scanning including:

    - Scanning protocol, including scanning hardware and software (e.g., make, model number of version), and scan parameters, including laser power, spatial resolution, pixel space and photomultiplier tube (PMT) voltage.
    - Scanned images. There is no consensus in MGED as to whether the images themselves should be provided. There are two advantages of providing images. First they are the raw data, and thus provide better validation of results, particularly where features may be flagged. Second, advances in feature extraction software may mean that it would be desirable to revisit old images and obtain new quantitative data. However, images are large in size and so inclusion of images would be expensive and difficult for many laboratories.
- *Image analysis and quantitation*.
    - Image analysis software. The specification and version of the feature extraction software, the algorithm and all parameters used.
    - Image analysis output. For each image, the complete output of the image analysis software. This is image quantitation table.
- *Normalized and summarized data*. This is gene expression data matrix containing data from the whole experiment.
    - Data processing protocol, including details of any normalization algorithms used.
- *Gene expression data tables*:
    - Derived measurement values. These summarize the replicated (whether on the same or different arrays), or different elements (sequences) for the same gene.
    - Reliability indicator for each data point, e.g. standard deviation or median absolute deviation. The inclusion of a reliability indicator is strongly encouraged but not essential.

Appendix B. Biology Glossary

Allele	*An alternative form of a gene or any other segment of a chromosome.*
cDNA	*Complementary DNA. A DNA copy of an mRNA or complex sample of mRNAs, made using reverse transcriptase.*
Chemical compound	*A distinct and pure substance formed by the union or two or more elements in definite proportion by weight.*
Codon	*A sequence of three nucleotides in messenger mRNA that specifies an amino acid.*
Consensus sequence	*A derived nucleotide sequence that represents a family of similar sequences. Each base in the consensus sequence corresponds to the base most frequently occuring at that position, in the real sequences.*
Contig	*A contiguous region of DNA sequence constructed by aligning many sequence "reads" (one "read" is the data generated from one sequencing reaction).*
EST	*Expressed Sequence Tag. A partial sequence of a randomly chosen cDNA, obtained from the results of a single DNA sequencing reaction. ESTs are used both to identify transcribed regions in genomic sequence and to characterize patterns of gene expression in the tissue that was the source of the cDNA.*
Exon	*Part of a gene that can encode amino acids in a protein. Usually adjacent to a non-coding DNA segment called an intron.*
Gene Expression	*The process by which a gene's coded information is translated into the structures present and operating in the cell (either proteins or RNAs).*
Gene Regulatory Networks	*The on-off switches and rheostats of a cell operating at the gene level. They dynamically orchestrate the level of expression for each gene in the genome by controlling whether and how vigorously that gene will be transcribed into RNA.*
Histone	*A basic protein associated with nucleic acids. Histones are important parts of the DNA control system, suppressing the expression of or causing the expression of specific parts of the DNA blueprints in conjunction with other nucleoproteins.*
Homologous Sequence	*In phylogenetics, describing particular features in different individuals that are genetically descended from the same feature in a common ancestor. In molecular biology, homologous sequences often mean significantly similar sequences that are highly likely to have a common descent.*
Kinase	*An enzyme that is important in regulating cell functions.*
mRNA	*Messenger RNA; arises in the process of transcription from the DNA and includes information on the synthesis of a protein.*
Nucleic Acid	*A biological molecule composed of a long chain of nucleotides. DNA is made of thousands of four different nucleotides repeated randomly*
Nucleotide	*Building blocks of DNA and RNA. Nucleotides are composed of phosphate, sugar and one of four bases, adenine, guanine, cytosine and uracil (RNA) or thymine (DNA). Three bases form a codon, which specifies a particular amino acid; amino acids are strung together to form proteins. Strings of thousands of nucleotides form a DNA or RNA molecule.*
Oligo	*Same as oligonucleotide.*
Oligomer	*A molecule containing a small number of covalently linked units; a multisubunit protein.*
Oligonucleotide	*A molecule usually composed of 25 or fewer nucleotides; used as a DNA synthesis primer.*
ORF	*Open Reading Frame. A section of a sequenced piece of DNA that begins with an initiation (methionine ATG) codon and ends with a nonsense codon. ORFs all have the potential to encode a protein or polypeptide, however many may not actually do so.*

Promoter	*A DNA sequence that is located in front of a gene and controls gene expression. Promoters are required for binding of RNA polymerase to initiate transcription.*
Ribosome	*Organelle of the cell. It walks down the messenger RNA three nucleotides at a time, building a new protein piece-by-piece. It has its own DNA (ribosomal DNA) and proteins (Ribosomal Proteins).*
SNP	*Single Nucleotide Polymorphism. A SNP (pronounced "snip") is a place in the genetic code where DNA differs from one person to the next by a single letter. These slight genetic variations between human beings may predispose some people to disease and explain why some respond better to certain drugs.*
Splice Variants	*A gene has splice variants if the organism can make different transcripts of the gene by using different exons. It is thought that many genes from eukaryotic organisms have splice variants. The different splice variants of a gene have different sequences.*
Transcription	*The process of copying information from DNA into new strands of messenger RNA (mRNA). The mRNA then carries this information to the cytoplasm, where it serves as the blueprint for the manufacture of a specific protein.*