MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

IGOR TRAJKOVSKI

FUNCTIONAL INTERPRETATION OF GENE EXPRESSION DATA

DOCTORAL DISSERTATION

LJUBLJANA, DECEMBER 2007

FUNCTIONAL INTERPRETATION OF GENE EXPRESSION DATA

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia, December 2007

Supervisor: Professor Nada Lavrač, Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:

Professor Sašo Džeroski, Chairman, Jožef Stefan Institute, Ljubljana, Slovenia
Professor Filip Železný, Member, Czech Technical University in Prague, Prague, Czech Republic
Professor Ljupčo Todorovski, Member, University of Ljubljana, Ljubljana, Slovenia

Igor Trajkovski

FUNCTIONAL INTERPRETATION OF GENE EXPRESSION DATA

Doctoral dissertation

FUNKCIJSKA INTERPRETACIJA PODATKOV O IZRAŽENOSTI GENOV

Doktorska disertacija

Supervisor: Professor Nada Lavrač

December 2007

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL
Ljubljana, Slovenia

I dedicate this thesis to my parents
Stevan and Zorica, and my sister Daniela

Preface

On 26 June, 2000, the sciences of biology and medicine changed forever. Prime Minister of the United Kingdom Tony Blair and President of the United States Bill Clinton held a joint press conference, linked via satellite, to announce the completion of the draft of the Human Genome. The New York Times ran a banner headline: 'Genetic Code of Human Life is Cracked by Scientists'. The sequence of three billion bases was the culmination of over a decade of work, during which the goal was always clearly in sight and the only questions were how fast the technology could progress and how generously the funding would flow.

The human genome is only one of the many complete genome sequences known. Taken together, genome sequences from organisms distributed widely among the branches of the tree of life give us a sense, only hinted at before, of the very great unity in detail of all life on Earth. They have changed our perceptions, much as the first pictures of the Earth from space presented a unified view of our planet.

The sequencing of the human genome sequence ranks with the Manhattan project that produced atomic weapons during the Second World War, and the space program that sent people to the Moon, as one of the great bursts of technological achievement of the last century. These projects share grounding in fundamental science, and large-scale and expensive engineering development and support. For biology, neither the attitudes nor the budgets will ever be the same.

The human genome is fundamentally about information, and computers were essential both for the determination of the sequence and for the applications to biology and medicine that are already resulting from it. Computing contributed not only the raw capacity for processing and storage of data, but also the mathematically-sophisticated methods required to achieve the results. The marriage of biology and computer science has created a new field called bioinformatics. Today bioinformatics is an applied science, where we use computer programs to make inferences from the data archives of modern molecular biology, to make connections among them, and to derive useful and interesting predictions.

Landmarks in the Human Genome Project

1953	Watson-Crick structure of DNA published.
1975	F. Sanger, and independently A. Maxam and W. Gilbert, develop methods for sequencing DNA.
1977	Bacteriophage ϕX-174 sequenced: first 'complete genome.'
1980	US Supreme Court holds that genetically-modified bacteria are patentable. This decision was the original basis for patenting of genes.
1981	Human mitochondrial DNA sequenced: 16 569 base pairs.
1984	Epstein-Barr virus genome sequenced: 172 281 base pairs.
1990	International Human Genome Project launched - target horizon 15 years.
1991	J. C. Venter and colleagues identify active genes via Expressed Sequence Tags - sequences of initial portions of DNA complementary to messenger RNA.
1992	Complete low resolution linkage map of the human genome.
1992	Beginning of the Caenorhabditis elegans sequencing project.
1992	Wellcome Trust and United Kingdom Medical Research Council establish The Sanger Centre for large-scale genomic sequencing, directed by J. Sulston.
1992	J. C. Venter forms The Institute for Genome Research (TIGR), associated with plans to exploit sequencing commercially through gene identification and drug discovery.
1995	First complete sequence of a bacterial genome, Haemophilus influenzae, by TIGR.
1996	High-resolution map of human genome - markers spaced by $\approx$ 600 000 base pairs.
1996	Completion of yeast genome, first eukaryotic genome sequence.
1998	Celera claims to be able to finish human genome by 2001. Wellcome responds by increasing funding to Sanger Centre.
1998	Caenorhabditis elegans sequence published.
1999	Drosophila melanogaster genome sequence announced, by Celera;
1999	Human Genome Project states goal: working draft of human genome by 2001 (90% of genes sequenced to >95% accuracy).
1999	Sequence of first complete human chromosome published.
2000	Joint announcement of complete draft sequence of human genome.
2003	Fiftieth anniversary of discovery of the structure of DNA. Completion of high-quality human genome sequence by public consortium.

Bioinformatics, however, continues to evolve very rapidly. In the past eight years, full genome sequencing has initiated the development of several high-throughput technologies, such as DNA microarrays and mass spectrometry, which have considerably progressed. These high-throughput technologies are capable of rapidly producing terabytes of data that are too overwhelming for conventional biological approaches and as a result, the need for computer/statistical/machine learning techniques for their processing and interpretation is today *stronger* rather than weaker.

Large databases of biological information created both challenging data mining problems but also a lot of opportunities, each requiring new ideas. In this regard, conventional computer science algorithms have been useful, but are increasingly unable to address many of the most interesting gene expression analysis problems. This is due to the inherent complexity of biological systems, brought about by evolutionary randomness, and to our lack of a comprehensive theory of life's organization at the molecular level. Machine-learning approaches (e.g., learning of decision trees, neural networks, bayesian models,

support vector machines, etc.), on the other hand, are ideally suited for domains characterized by the presence of large amounts of data, 'noisy' patterns, and the absence of general theories. The fundamental idea behind these approaches is to learn the theory *automatically* from the data, through a process of inference, model fitting, or learning from examples. Thus they form a viable complementary approach to conventional methods.

Modeling biological data probabilistically really makes sense. One reason is that biological measurements are often inherently 'noisy', as is the case of DNA microarray or mass spectrometer data. However, measurement noise can not be the sole reason for modeling biological data probabilistically. The real need for modeling biological data probabilistically comes from the complexity and variability of biological systems brought about by eons of evolutionary experimentation in complex environments. As a result, biological systems have inherently a very high dimensionality. Even in microarray experiments where expression levels of thousands of genes are measured simultaneously, only a small subset of the relevant variables is being observed. The majority of the variables remains 'hidden' and must be factored out through probabilistic modeling.

It is the merging of all three factors: easily accessible biological data, computers and theoretical probabilistic framework that is fueling the machine learning and data mining expansion in bioinformatics and elsewhere. And it is fair to say that bioinformatics and machine learning methods have started to have a significant impact on biology and medicine.

Biology has traditionally been an observational rather than a deductive science. Although recent developments have not altered this basic orientation, the nature of the data has radically changed. It is arguable that until recently all biological observations were fundamentally anecdotal - admittedly with varying degrees of precision, some very high indeed. However, in the last generation the data have become not only much more quantitative and precise, but, in the case of nucleotide and amino acid sequences, they have become discrete. It is possible to determine the genome sequence of an individual organism not only completely, but in principle exactly. Experimental error can never be avoided entirely, but for modern genomic sequencing it is extremely low.

Note that this has converted biology into a deductive science. Life does obey principles of physics and chemistry, but for now life is too complex for us to deduce its detailed properties from the basic principles.

A second obvious property of these data is their *very large amount*. Currently the nucleotide sequence databanks contain 10^{11} bases (abbreviated 100 Gbp). If we use the approximate size of the human genome - 3.2×10^9 letters - as a

unit, this amounts to thirty HUman Genome Equivalents (or 30 *huges*). For a comprehensible standard of comparison, 1 *huge* is comparable to the number of characters appearing in six complete years of issues of *The New York Times*. The database of macromolecular structures contains 42,000 entries, the full three-dimensional coordinates of proteins, of average length 400 residues. Not only are the individual databanks large, but their sizes are increasing at a very high rate.

The quality and quantity of these data have encouraged scientists to set equally ambitious goals:

- To be able to say: "We saw life clearly and saw it whole". That is, to understand integrative aspects of the biology of organisms, viewed as coherent complex systems.
- To interrelate sequence, three-dimensional structure, expression data, interactions and functions of individual genes.
- To use data on contemporary organisms as a basis for a travel backward and forward in time - back to deduce events in evolutionary history, forward to greater careful scientific modification of biological systems.
- To support applications to biology, medicine, agriculture and other scientific fields.

Our ability to achieve these goals in the future will strongly depend on our ability to combine and correlate diverse datasets along multiple dimensions and scales, and progressively switch from the accumulation of data to its *interpretation*. *Gene expression data* will have to be integrated with structural, functional, pathway, phenotypic and clinical data, and so forth. Basic research within bioinformatics will have to deal with these issues of system and integrative biology, in the situation where the amount of data is growing exponentially[1].

This thesis contributes to the development of methods for the interpretation of high-throughput DNA microarray data, by integrating and using various sources of biological data (gene ontologies, gene annotations and gene-gene interactions) with the goal of extracting new biological knowledge hidden in this abundance of data.

[1]The content of the preface was kindly borrowed from the books *Introduction to Bioinformatics* (2002), Arthur M. Lesk and *Bioinformatics, The Machine Learning Approach* (1998), Pierre Baldi.

Contents

Preface iii

Abstract 5

Povzetek 7

1 Introduction 9
- 1.1 Motivation . . . 9
- 1.2 Problem statement . . . 10
- 1.3 Hypothesis . . . 11
- 1.4 Relational approach to functional interpretation of gene expression data . 13
- 1.5 Scientific contributions . . . 14
- 1.6 Organization of the thesis . . . 15

2 Gene Expression Data and Gene Ontologies 17
- 2.1 Cellular biology and gene regulation . . . 17
- 2.2 Techniques for measuring gene expression . . . 19
- 2.3 Gene expression data analysis . . . 22
  - 2.3.1 Preprocessing . . . 22
  - 2.3.2 High-level analysis . . . 24
  - 2.3.3 Further analysis . . . 29
- 2.4 Gene Ontologies . . . 30
  - 2.4.1 Ontology design and implementation . . . 31
  - 2.4.2 Three ontologies of GO . . . 32
  - 2.4.3 Gene annotations . . . 34
  - 2.4.4 Biological pathways . . . 35

3 Functional Interpretation of Gene Expression Data 41
- 3.1 Threshold-based functional interpretation . . . 42
  - 3.1.1 Fisher's exact test . . . 43
  - 3.1.2 Statistical approaches to test significant biological differences . . 44
  - 3.1.3 Multiple testing . . . 46

3.2 Threshold-free functional interpretation . . . 51
3.2.1 Gene Set Enrichment Analysis (GSEA) . . . 53
3.2.2 Parametric Analysis of Gene set Enrichment (PAGE) . . . 55
3.3 Discussion . . . 56

4 Construction of an Integrated Database 59
4.1 Integration of GO and KEGG Orthology . . . 60
4.2 Integration of GO and KO gene annotations . . . 61
4.3 Gene-gene interaction data . . . 63
4.4 Gene expression data . . . 65

5 Learning Relational Descriptions of Differentially Expressed Gene Sets 67
5.1 Related work . . . 68
5.2 Descriptive analysis using relational features . . . 68
5.2.1 The RSD algorithm . . . 71
5.3 Experiments . . . 77
5.3.1 Materials and methods . . . 77
5.3.2 Experimental results . . . 80
5.3.3 Statistical validation . . . 81
5.3.4 Analyzing individual components of the methodology . . . 83
5.4 Discussion . . . 84

6 SEGS: Search for Enriched Gene Sets 87
6.1 Related work . . . 88
6.2 The proposed SEGS approach . . . 89
6.2.1 Properties of GO and KO terms . . . 90
6.2.2 Basic SEGS operators for gene set construction using GO, KO and ENTREZ . . . 90
6.2.3 Pruning the search space for enriched gene sets . . . 93
6.3 Experiments . . . 93
6.3.1 Brief description of datasets . . . 95
6.3.2 Experimental results . . . 95
6.3.3 Statistical validation . . . 95
6.3.4 Biomedical significance of the discovered enriched gene sets . . . 98
6.4 Discussion . . . 102

7 Conclusions and Further Work 103

Acknowledgement 107

References 109

List of Figures **119**

List of Tables **121**

Extended Abstract **123**

Biography **129**

Abstract

Microarrays are at the center of a revolution in biotechnology, allowing researchers to simultaneously monitor the expression of tens of thousands of genes. The final aim of a typical microarray experiment is to find a molecular explanation for a given macroscopic observation (e.g., which pathways are affected by the loss of glucose in a cell, what biological processes differentiate a healthy control from a diseased case); this is called *functional interpretation* of gene expression data.

This thesis presents two new methods for the functional interpretation of gene expression data that combine and use knowledge stored in different kinds of biological databases. The interpretation is done by identifying and describing gene sets that have significantly altered expression profile (e.g., over- or under-expressed). The search of the interesting gene sets is performed in the space of already defined gene sets (genes that have common annotation by predefined ontological terms) and in the space of newly generated gene sets that have predefined characteristics (e.g., the minimum number of member genes that are found to be differentially expressed). Three well established methods, Fisher's exact test, Gene Set Enrichment Analysis (GSEA), and Parametric Analysis of Gene set Enrichment (PAGE), were employed in order to identify gene sets with significantly altered expression profiles.

Both developed methods share the same mechanism of first-order (relational) feature construction, by using the *Gene Ontology (GO), Kyoto Encyclopedia of Genes and Genomes (KEGG) Orthology, gene annotations*, and *gene-gene interaction data*. These features, constructed by the propositionalization mechanism of the Relational Subgroup Discovery algorithm (RSD), are used as generalized gene annotations.

The first method belongs to the class of threshold-based functional analysis methods. It is performed in two steps. In the first step, 'top' genes of interest are *selected* using gene differential expression as a selection criterion. The selection process does not take into account the fact that gene products are acting cooperatively in the cell and consequently, for better interpretation of the selected gene list, in the second step their behavior must be coupled to some extent by looking for their common description. The language used for describing the functionality of the genes is constructed from GO, gene annotations, and gene-gene interaction data. By using this background knowledge together with the paradigm of relational subgroup discovery we found common descriptions of gene sets differentially expressed in specific cancers. The

descriptions of these gene sets can be straightforwardly used by the medical experts.

The second method is based on threshold-free functional analysis. This method is also performed in two steps. In the first step, genes are *ranked* by using their differential expression values when comparing predefined classes (e.g., tumor vs. healthy controls) by means of an appropriate statistical test (e.g., the t-test). In the second step, the positions of the members of the predefined gene sets (e.g., defined by GO and KEGG Orthology terms) in the ranked list are analyzed using appropriate statistical tests (e.g., the Kolmogorov-Smirnov test). Gene sets, whose members are predominantly found at the top of the list, are considered enriched and responsible for the phenotype difference (e.g., the tumor vs. normal). Our contribution to this methodology is a development of an *efficient algorithm*, inspired by the RSD first-order features construction, for the construction of *new*, potentially enriched, gene sets. New gene sets are defined by conjunctions of relational features constructed from the background knowledge.

The two developed methods have proved to be of interest to medical experts. The extracted knowledge turns out to be consistent with the relevant literature, and proves to have the potential for guiding the biomedical research and generating new hypotheses that explain microarray measurements.

Also, a by-product of the thesis is an easy to use relational database that integrates several sources of biological knowledge (GO, KEGG Orthology, gene annotations and gene-gene interaction data) in a unified format. This database is now publicly available to a wider scientific community.

Povzetek

Genske mikromreže so v žarišču biotehnološke revolucije saj omogočajo sočasno merjenje izraženosti več deset tisoč genov. Cilj tipičnega eksperimenta z mikromrežami je najti *funkcijsko interpretacijo* izraženosti genov, z drugimi besedami molekularno razlago za makroskopska opažanja (npr. na katere poti vpliva zmanjšanje glukoze v celici, kateri biološki proces je pomemben za razlikovanje med zdravimi in bolnimi primerki, ipd).

V doktorski disertaciji predstavimo dve novi metodi za funkcijsko interpretacijo podatkov o izraženosti genov. V obeh primerih poleg podatkov o izraženosti genov uporabimo še biološko znanje, ki je shranjeno v različnih podatkovnih bazah. Interpretacijo naredimo tako, da identificiramo in opišemo gene, ki imajo signifikantno spremenjeno izraženost profila (npr. tiste, ki so nadpovprečno- ali podpovprečno- izraženi). Zanimive množice genov iščemo med že definiranimi množicami genov (to so geni, ki imajo skupno anotacijo v ontologiji) in med na novo generiranimi množicami genov, ki imajo v naprej definirane značilnosti (npr. minimalno število diferencialno izraženih genov v množici). Uporabili smo tri uveljavljene metode za identifikacijo množic genov s signifikantno spremenjenim profilom izražanja: Fišerjev test (Fisher's exact test), Gene Set Enrichment Analysis (GSEA) in Parametric Analysis of Gene set Enrichment (PAGE).

Obe razviti metodi uporabljata isti mehanizem za gradnjo relacijskih značilk z uporabo ontologije genov GO (Gene Ontology), enciklopedije genov in ortologije genomov KEGG (Kyoto Encyclopedia of Genes and Genomes Orthology), anotacije genov in podatkov o interakciji med geni. Značilke zgrajene s postopkom propozicionalizacije algoritma RSD (Relational Subgroup Discovery) uporabimo kot posplošene anotacije genov.

Prva metoda temelji na funkcijski analizi z omejevanjem. Izvaja se v dveh korakih: v prvem koraku *izberemo* 'najzanimivejše' gene glede na kriterij diferencialne izraženosti. Ker ta postopek izbire ne upošteva sodelovanja genov v celici jih v drugem koraku zaradi boljše interpretabilnosti združimo glede na njihove skupne opise. Jezik opisov za opisovanje funkcionalnosti genov je sestavljen iz GO, anotacij genov in podatkov o interakciji med geni. Z uporabo tega predznanja in paradigme relacijskega odkrivanja podskupin, implementirane v algoritmu RSD, smo našli opise skupin genov, ki so diferencialno izražene pri določenih tumorjih. To znanje lahko zdravniki direktno uporabijo.

Druga metoda temelji na funkcijski analizi brez omejevana. Tudi ta se izvaja v dveh korakih: v prvem koraku gene z uporabo primernega statističnega testa (npr. t-test) *razvrstimo* glede na njihovo diferencialno izraženost v vnaprej

določenih razredih (npr. tumor v primerjavi z zdravim tkivom). V drugem koraku analiziramo pozicije elementov množic genov (množice genov definiramo npr. kot terme v GO ali KEGG) v razvrstitvi dobljeni z uporabo primernega statističnega testa (npr. Kolmogorov-Smirnov test). Množice genov, katerih elementi so večinoma v začetku razvrstitve, so obogatene in odgovorne za fenotipsko razlikovanje (npr. tumorja v primerjavi z zdravim tkivom). Naš prispevek k tej metodologiji je razvoj *učinkovitega algoritma* za gradnjo novih - možno obogatenih - množic genov. Iz predznanja sestavljamo opise množic genov kot konjunkcije relacijskih značilk po vzoru gradnje relacijskih logičnih značilk algoritma RSD.

Ti dve metodi sta potencialno zanimivi za zdravnike. Izkazalo se je namreč, da je avtomatsko izluščeno znanje skladno z relevantno literaturo tega področja in da ima potencial za usmerjanje biomedicinskih raziskav s tega področja in za generiranje novih hipotez, ki razlagajo eksperimente z mikromrežami.

Poleg naštetega je rezultat disertacije tudi uporabniško prijazna podatkovna baza, ki združuje več bioloških virov podatkov (GO, KEGG Orthology, anotacije genov in podatke o interakciji genov) v enotnem formatu. Ta baza je zdaj javno dostopna širši znanstveni skupnosti.

1 Introduction

This chapter presents the motivation for the work in this thesis, a short introduction to the problem of functional interpretation of gene expression data, the hypothesis that we prove in the thesis and a list of specific scientific contributions of our work.

1.1 Motivation

Can we live forever ? Can we stop the process of aging by changing the information processes underlying the biology ?

An answer to these questions can be given by the following metaphor of maintaining a house. How long does a house last? The answer obviously depends on how well you take care of it. If you do not repairs, the roof will leak, so water and the elements will invade, and eventually the house will disintegrate. But if you proactively take care of the structure, repair all damage, confront all dangers, and rebuild or renovate parts from time to time using new materials and technologies, the life of the house can essentially be extended without limit.

The same holds true for our bodies and brains. The only difference is that, while we fully understand the methods underlying the maintenance of a house, we do not yet fully understand all of the biological principles of life. But with our rapidly increasing comprehension of the biological processes and pathways of biology, we are quickly gaining this knowledge. We are beginning to understand that aging and diseases are not one single unstoppable progression process, but a group of related processes. Strategies are emerging for fully reversing each of these aging and disease progressions, using different combinations of biotechnology methods.

These questions are the main motivation for doing this work, improving the existing and developing new methods for automatic knowledge discovery concerning biological processes and molecular functions that govern specific diseases.

One powerful approach to start the investigation of these processes are high-throughput techniques, such as DNA microarrays, that measure expression of genes. Gene expression is the process by which specific cellular components produce proteins according to a specific genetic blueprint, subsequence of DNA, or *gene*. With recently developed gene technologies we are on the verge of being able to control how genes express themselves. Many new therapies now in development and testing are based on manipulating gene expression, by either turning off the expression of disease-causing genes or by turning on desirable genes that may otherwise not be expressed in a particular type of cell.

1.2 Problem statement

Over the past few years, due to the popularization of high-throughput techniques, the possibility of obtaining experimental data has increased significantly. Nevertheless, the *functional interpretation* of the results, which involves translating these data into useful biological knowledge, still remains a challenge.

The aim of the methods for the functional interpretation of microarray experiments is to find a functional explanation at molecular level that accounts for the macroscopic observation related to the hypothesis that originated the experiment (e.g., why a number of genes are responsible for the physiological differences between healthy and diseased people). This is achieved through the study of the over-representation of some type of functionally relevant labels in the genes detected as important in the experiment (see Figure 1.1).

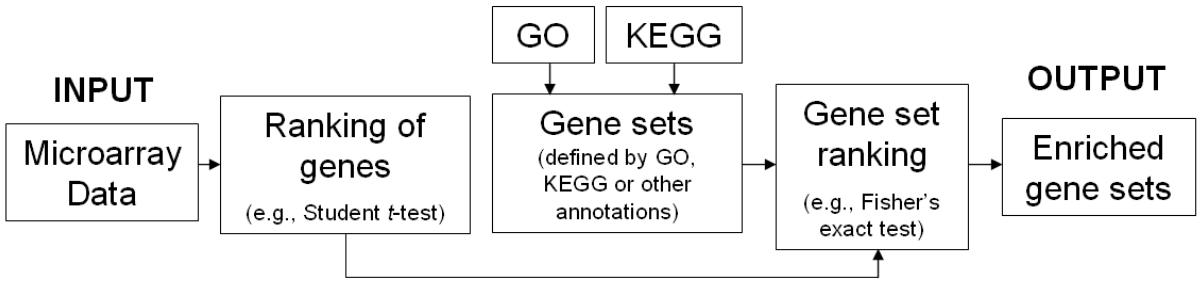


Figure 1.1: Data flow of a typical functional interpretation of gene expression data.

Two main approaches are currently in use: threshold-based and threshold-free. In the first case, the conclusions are reached by means of a two-steps process where the important genes are firstly selected based upon their experimental values (e.g., using a test for differential expression between two classes or a clustering method for finding co-expressed genes, etc.) Then, this selection is analyzed for the significant enrichment of biological terms with functional meaning (e.g., Gene Ontology (GO) (6) or Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways (42)) using different tests (e.g., Fisher's exact test) (45). Programmes such as OntoExpress (25), FatiGO (1), GOMiner (90), etc., can be considered as representatives of a family of methods that use these terms to find clues for the interpretation of the results of microarray experiments (45). By means of this simple two-step approach, a reasonable biological functional interpretation of a microarray experiment can be attained.

Several authors have pointed out that the first step of such a strategy, where genes are selected without taking into account their cooperative behavior, would constitute its Achile's heel. If the genes are considered as independent and tested one at a time, then

very rigorous thresholds need to be used to reduce the rate of false positives (1). The obvious consequence of this is the reduction of the sensitivity in the second step.

That is one of the reasons that initiated the development of a new generation of procedures which draw inspiration from molecular systems biology. Threshold-free methods avoid the first step of selection by not considering genes alone, but in functional blocks. These methods have proved to be much more sensitive than the threshold-based alternatives. The Gene Set Enrichment Analysis (GSEA) (75) and Parametric Analysis of Gene Set Enrichment (PAGE) (47) have pioneered a family of methods devised not to find individual genes but to search for groups of functionally related genes with a coordinate (although not necessarily high) over- or under-expression across a list of genes ranked by their differential expression between classes of microarray data. With this aim in mind different tests have recently been proposed for analyzing microarray data (2; 31).

Moreover, from the point of view of systems biology, threshold-free methods are far more consistent because they directly test pre-defined functionally-related blocks of genes. These blocks are formed by genes that share functional labels, which are supposed to account for the cooperative roles fulfilled by these genes in the cell. Such functionally-related blocks of genes would provide a molecular-level explanation for the macroscopic traits studied in the microarray experiment.

1.3 Hypothesis

Even with the introduction of new methods, very often after correcting for multiple hypotheses testing, few (or no) GO or KEGG terms turn out to meet the threshold for statistical significance, because the relevant biological differences are small relative to the noise inherent to the microarray technology.

In this thesis we try to solve this problem by constructing new gene sets using the existing gene sets and utilizing the gene-gene interaction data. These newly constructed gene sets are later tested for possible enrichment (i.e., checking if the gene set is significantly over-represented in the selected important genes, or if it shows collective over-expression across a list of genes ranked by their differential expression).

The construction of the new gene sets is based on the mechanism of first-order (relational) features construction, by using GO, KEGG, gene annotations and gene-gene interaction data. These features are used as generalized gene annotations. For example, in addition to the usual gene features derived from GO and KEGG:

```
gene_feature_1(X) = function(X, 'DNA_Binding')
```

which is used to annotate all genes that perform molecular function 'DNA_Binding', we define features like this one:

```
gene_feature_2(X) = interaction(X, Y), function(Y, 'DNA_Binding')
```

which is used to annotate all genes that interact with genes that perform molecular function 'DNA_Binding', or

```
gene_feature_3(X) = function(X, 'DNA_Binding'), component(X, 'nucleus')
```

which is used to annotate all genes that perform molecular function 'DNA_Binding' and operate in the 'nucleus'.

The construction of this kind of gene sets really makes sense. An increasing corpus of evidence reveals that genes do not operate alone within the cell, but in an intricate network of interactions that we only recently started to discover (35; 67; 73). It is widely accepted that co-expressing genes tend to fulfill common roles in the cell (53; 74), and in fact, this causal relationship has been used to predict gene function from patterns of co-expression (59; 81). This clearly shows the necessity for methods and tools to aid the functional interpretation of large-scale experiments such as microarrays, and to formulate genome-scale hypotheses from a systems biology perspective, in such a way that the collective properties of groups of genes are taken into account. Therefore, adding the interacting properties of the genes will greatly improve the functional interpretation of gene expression data.

Second, very often there are scenarios when not all genes annotated by some GO term are over-expressed, but only a subgroup of them. Allowing conjunctions of gene features is an elegant way to make them more specific and more useful, while retaining their comprehensibility. For example, if we test the enrichment of the gene set defined by GO term 'nucleus', we can not expect that all genes in the nucleus will be over-expressed. The same holds for GO term 'transport', there are thousands of genes operating inside and outside the cell that perform this function. When testing this gene set for enrichment we will probably find that it is not enriched. But, if we define a gene set whose member genes are annotated by both GO terms, 'nucleus' and 'transport', there is a bigger chance that we will discover the over-expressed genes which transport molecules inside(outside) the nucleus.

1.4 Relational approach to functional interpretation of gene expression data

The main novelty of this thesis is the relational approach to functional interpretation of gene expression data by using the newly constructed features which have enabled the development of the following two methods.

The first method (see Chapter 5) belongs to the class of threshold-based functional analysis methods. It is performed in two steps. In the first step, 'top' genes of interest are selected using gene differential expression as a selection criterion. In the second step their behavior is coupled to some extent by looking for their common description. Relational features used for describing the functionality of the genes (described in Section 1.3) are constructed from GO, gene annotations, and gene-gene interaction data. By using these features together with the paradigm of relational subgroup discovery we found common descriptions of gene sets differentially expressed in specific cancers. The descriptions of these gene sets can be straightforwardly used by the medical experts (see Section 5.3.2).

The second method (see Chapter 6) is based on threshold-free functional analysis. This method is also performed in two steps. In the first step, genes are ranked by using their differential expression values when comparing predefined classes (e.g., tumor vs. healthy controls) by means of a appropriate statistical test (e.g., the t-test). In the second step, the positions of the members of the predefined gene sets (e.g., defined by GO and KEGG Orthology (KO) terms) in the ranked list are analyzed using appropriate statistical tests (e.g., the Kolmogorov-Smirnov test). Gene sets, whose members are predominantly found at the top of the list, are considered enriched and responsible for the phenotype difference (e.g., the tumor vs. normal). Our contribution to this methodology, inspired by relational feature construction, is a development of an *efficient algorithm* (that uses the GO and KO topology) for the construction of new, potentially enriched (collectively differentially expressed), gene sets. The experimental results show that the introduced method improves the functional interpretation of gene expression data (see Section 6.3.2). We base our conclusion on the following facts: Enrichment scores of the newly constructed sets are better then the enrichment scores of any single GO and KO term, and newly constructed enriched gene sets are sometimes described by non-enriched GO and KO terms, which means that we are extracting additional biological knowledge that can not be found by single term enrichment analysis.

1.5 Scientific contributions

This thesis contributes to the fields of Bioinformatics and Data mining.
The specific contributions include:

- A state-of-the-art overview of gene expression data analysis, with focus on functional interpretation of gene expression data, presented in Chapters 2 and 3.
- The development of an easy to use, easily updateable relational database, integrating four different sources of information (GO, KEGG, gene annotations and gene-gene interaction data). This database, described in Chapter 4, is made publicly available to other researchers.
- The development of a new methodology for learning relational descriptions of differentially expressed gene sets using the integrated background knowledge and the methodology of relational subgroup discovery, described in Chapter 5.
- The development of a new method for the efficient construction of biologically relevant enriched gene sets, using the integrated background knowledge and various methods for gene set enrichment analysis, described in Chapter 6.

The main scientific contributions of this work were published in the following papers:

- Igor Trajkovski, Filip Železný, Jakub Tolar, Nada Lavrač: Relational Subgroup Discovery for Descriptive Analysis of Microarray Data. *Proceedings of the 2nd International Symposium on Computational Life Science (CompLife)* 2006: pp. 86-96, Cambridge, UK.
- Igor Trajkovski, Nada Lavrač: Efficient Generation of Biologically Relevant Enriched Gene Sets. *Proceedings of the International Symposium on Bioinformatics Research and Applications (ISBRA)* 2007: pp. 248-259, Atlanta, USA.
- Igor Trajkovski, Nada Lavrač: Interpreting Gene Expression Data by Searching for Enriched Gene Sets. *Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME)* 2007: pp. 144-148, Amsterdam, The Netherlands.
- Igor Trajkovski, Filip Železný, Nada Lavrač, Jakub Tolar: Learning Relational Descriptions of Differentially Expressed Gene Groups. *IEEE Transactions on Systems, Man, and Cybernetics, Special issue on Intelligent Computation for Bioinformatics*, accepted, to appear in January 2008.
- Igor Trajkovski, Nada Lavrač, Jakub Tolar: SEGS: Search for Enriched Gene Sets in Microarray Data. *Journal of Biomedical Informatics*, accepted in December 2007 for publication in 2008.

1.6 Organization of the thesis

This thesis is organized in the following way. Chapter 2 presents a state-of-the-art overview, providing the necessary background knowledge about the gene expression data and resources of biological knowledge stored in different types of databases, needed for understanding the existing methods for analyzing gene expression data and the new methods presented in this thesis. In Chapter 3 we review the work that relates to our functional interpretation of microarray data. Chapter 4 presents the process of collecting, preprocessing and uniformly formatting of publicly available biological databases and microarray datasets. Chapter 5 presents the first developed method for learning relational descriptions of differentially expressed gene groups. Chapter 6 presents the second developed method of searching for enriched gene sets in microarray data. At the end, in Chapter 7, we draw final conclusions and propose some directions for future work.

2 Gene Expression Data and Gene Ontologies

This chapter presents a state-of-the-art overview, providing the necessary background knowledge about the gene expression data: what these data represent, how they are measured and what preprocessing techniques for cleaning the data are used. It also presents several resources of biological knowledge stored in different types of databases used for functional interpretation of gene expression data.

2.1 Cellular biology and gene regulation

The cell is a complex machinery - a collection of organic molecules, intricately interacting, constituting what we may define as the basic unit of life. This microscopic mixture of molecules possesses an extraordinary ability to communicate with its environment and to regulate its own state according to internal and external stimuli.

The structural and functional building blocks of the cell are primarily *proteins*, but also *ribonucleic acids (RNA)*. Many of these molecules control reactions such as signaling and metabolism while others make up the skeleton and shell of the cell thus defining it spatially in its environment. The cell manufactures these molecules itself, relying on molecular blueprints that are inherited from cell to cell. The blueprints are stored in the *deoxyribonucleic acid (DNA)*, a molecule shaped as a double helix, where the two strands are joined together through pairs of *nucleotides*. It is the sequence of nucleotides that determines the information content in the DNA. The set of nucleotides in the DNA is made up of *adenine (A)*, *thymine (T)*, *guanine (G)* and *cytosine (C)*, so the information is encoded in a four-letter alphabet. The two strands of the double helix are complementary to each other since A always pairs with T and G always pairs with C. A *gene* is a portion of the DNA which contains the instructions for building a specific molecule, its *gene product*.

In principle, all cells in a given multicellular organism carry the same genetic code, identical to the one of the original fertilized egg. Nevertheless, higher order species consist of highly specialized cell types, appearing in different locations of the body, having different tasks. So why do skin cells, nerve cells and blood cells, which all have the same genetic code, behave so widely different? The answer is that different genes are active, or *expressed* in the different cell types, making them produce their own specific set of molecules.

The *protein synthesis*, that is, the process of producing a protein from the information in its corresponding gene can be divided into two phases - transcription and translation (Figure 2.1).

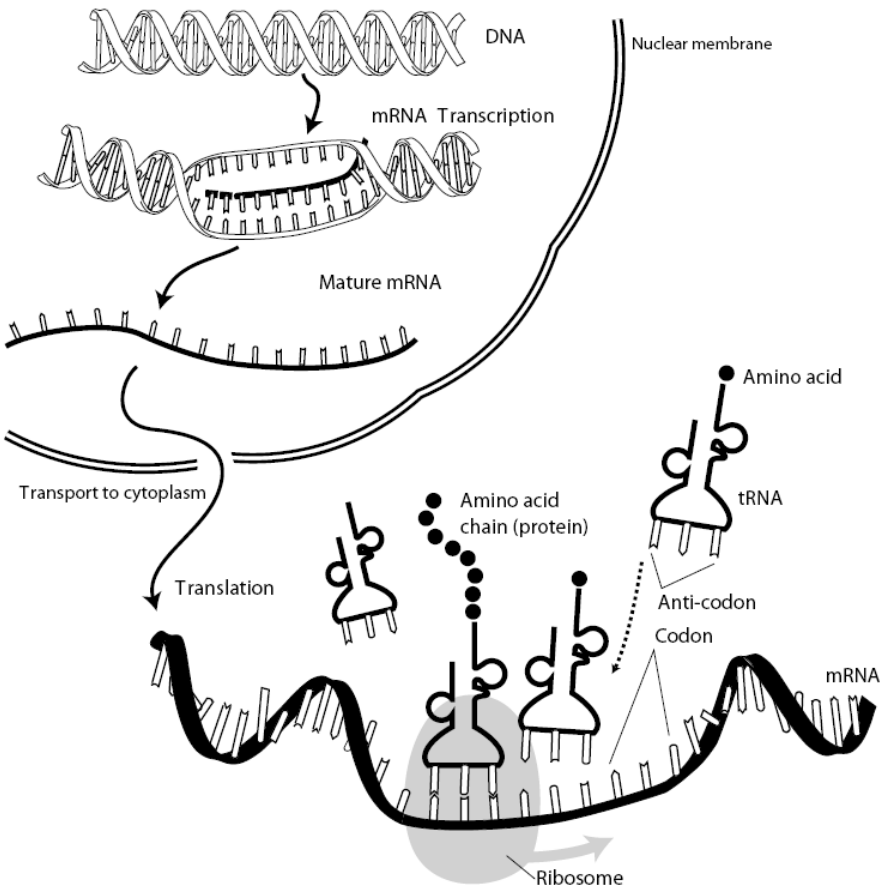


Figure 2.1: Schematic illustration of cells protein synthesis. The figure is printed by courtesy of the National Human Genome Research Institute, the National Institutes of Health.

During transcription, the genetic code of the gene is copied to a *messenger RNA (mRNA)* molecule, a single-stranded nucleic acid carrying the same nucleotides as DNA with the exception of thymine whose role is instead taken by *uracil (U)*. Transcription starts when the protein RNA polymerase binds to the *promoter region*, the start of the gene, and locally unzips the DNA helix so that the strands become free for reading. The RNA polymerase propagates along the strand while constructing an mRNA molecule by adding nucleotides complementary to those being passed by on the DNA molecule. Eventually, the RNA polymerase reaches a *terminator region* and stops transcribing, whereby the mRNA is released and the DNA resumes its double helix configuration. Following this, the primary mRNA is processed into mature mRNA by other molecules, for example by removing the parts corresponding to *introns*, non-coding regions of the DNA, in a process called *splicing*.

Following transcription, translation takes place, where the four-letter alphabet of the

DNA and mRNA is translated into the alphabet of proteins. Like the nucleic acids, proteins are polymers, albeit consisting of sequences of *amino acids* instead of nucleotides. The number of amino acids is 20 so the protein alphabet is one of 20 letters. In order to represent 20 amino acids with four nucleotides we need three nucleotides per amino acid. Such a three-nucleotide word is denoted a *codon* [1]. The actual translation between the two alphabets is accomplished by *transfer* RNA molecules (*tRNA*) which attach themselves to the mRNA. The tRNA has one end with a specific *anticodon*, that is, a complementary codon, and another end to which the corresponding amino acid is attached. The last step in the translation is performed by the ribosomes which join the sequence of amino acids found on the tRNA along the mRNA together to form the protein.

The production of RNA and proteins from a given gene does not take place independently of the expression of other genes. Conversely, gene products influence the production of other gene products using positive or negative feedback. This regulation is essential for the cell to be able to respond to internal and external circumstances and takes place at all levels in the chain of reactions that produce a protein from a gene sequence.

To enable transcription of a gene, the binding of certain proteins, *transcription factors*, to the DNA, is necessary. Different genes are either activated or repressed by different combinations of one or several transcription factors. Transcriptional control is not the only means of regulation. After transcription, mRNA molecules may interact with other gene products, resulting in altered structure or lifetime of the mRNA. After translation, subsequent protein-protein reactions may be required to finalize the functional protein.

The description above only sketches a few of the ways that genes interact to regulate each others expression. The main conclusion is that the cell can be viewed as a large dynamical system with different molecules interacting with each other. The explicit study of such *genetic regulatory networks* is often referred to as *systems biology* (although other wider definitions of the term are also used). Mathematical modeling of genetic regulatory networks had its principal breakthrough in the 1960's and various models have been proposed, ranging in complexity from discrete cellular automata models to detailed probabilistic models; see (23) for a review.

2.2 Techniques for measuring gene expression

Enabled by advances in measurement technology and the sequencing of genomes, such as the human in the Human Genome Project, the 1990's witnessed the emergence of technologies for global measurement of gene expression. Earlier techniques were limited to the study of a few genes at the time, while the *microarray* techniques gave biologists the tools to sample the expression of, in principle, the whole genome in one single measurement.

[1] Since $4^3 = 64 > 20$ there is a degeneracy in this representation; some amino acids are coded for by more than one codon. On the other hand, one codon codes for at most one amino acid.

Microarrays measure the abundance of mRNA from the set of genes at a given moment. From a *cell sample* of interest, mRNA is extracted and put in contact with an array on which *probes* (complementary sequences, or subsequences, of the genes) have been attached. The different mRNA in the solution then bind to their corresponding complements on the chip, and the amount of mRNA for each gene can be optically measured by a laser scanner.

There are two main microarray platforms currently in use; spotted microarrays (70) and high-density synthetic oligonucleotide[1] microarrays (56). These are basically two variations of the same general solution described above.

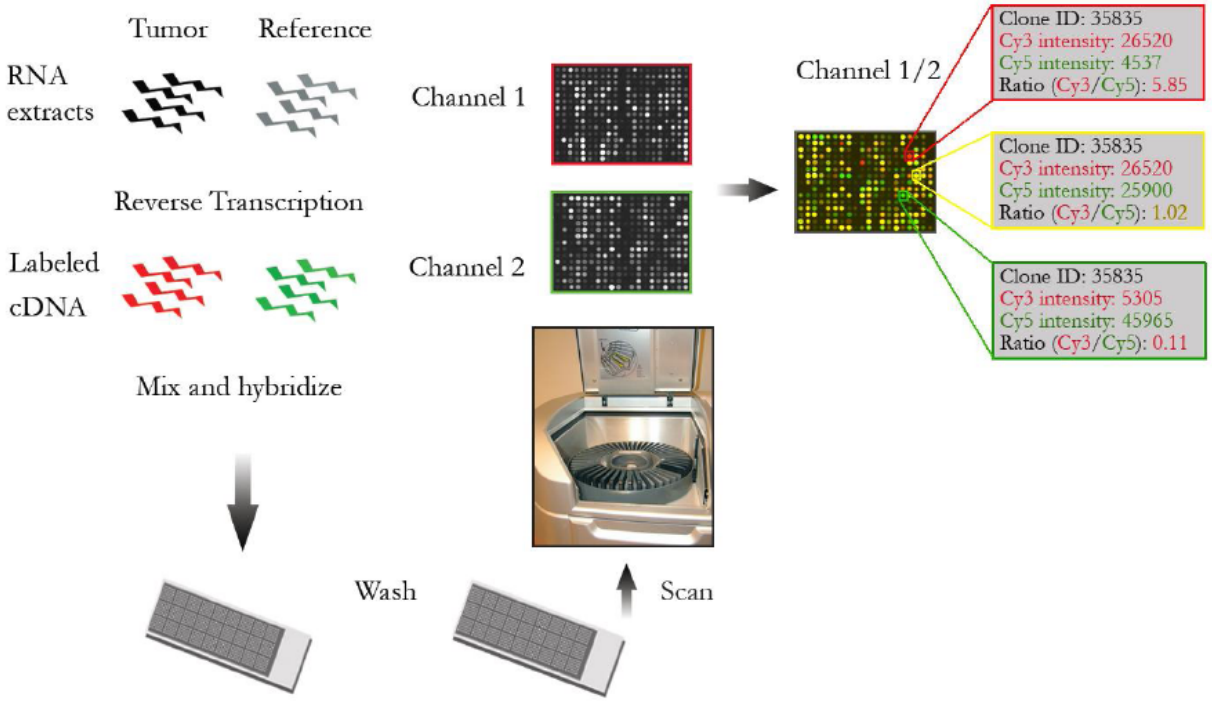


Figure 2.2: Spotted microarray technique. The figure is printed by courtesy of Anna Andersson, Department of Clinical Genetics, Lund University hospital.

A spotted microarray[2] (Figure 2.2) has probes consisting of cDNA or long oligo strands attached spot-wise on a glass slide in a grid shaped pattern. The platform is, in its most common form, a *two-channel technique*, meaning that in each measurement, the expression profiles of two cell samples are measured simultaneously.[3] After extracting RNA from the two samples it is reverse-transcribed to cDNA, and fluorescently labeled

[1] A short stretch of nucleotides, 2 to 200 nucleotides long.
[2] The well known cDNA microarray technique falls under the category of spotted microarrays.
[3] However, both one- and multi-channel spotted microarray platforms exist.

with Cy3 (green) for one sample and Cy5 (red) for the other. The cDNA molecules of the samples are denoted *targets*. After labeling, the two samples are mixed and put in contact with the probes on the slide. During *hybridization*, the targets bind to their corresponding probes, thus geometrically sorting the targets on the slide. Finally, each spot is illuminated by a laser at two different wavelengths; one yielding Cy3 fluorescence and one yielding Cy5 fluorescence. Thus two images are obtained; one with green spots and one with red spots, measuring the abundances of the respective sample targets. These images then go through a number of image processing steps. First, the spots need to be located and segmented out from the background. Second, the spot intensity and local background intensity is estimated from the pixels, commonly by taking the mean or the median of the pixel values. Thus for each spot, estimates of the red and green foreground and background intensities are available. For each channel, the expression level is estimated as

$$Y_i = FG_i - BG_i \tag{2.1}$$

where FG_i and BG_i are the foreground and background estimates at spot i for the particular channel. In principle, the two channels could be treated separately, but in multiarray experiments it is common practice to use a *reference sample*, common to all arrays, in one of the channels. The expression levels of the other sample, the *query sample*, are then reported as relative values compared to the reference expressions, i.e.,

$$Y_i = \frac{Y_i^{query}}{Y_i^{reference}} \tag{2.2}$$

Most commonly this ratio is subsequently transformed by taking the logarithm, as discussed in Section 2.3.1.1.

High-density synthetic oligonucleotide microarrays have a slightly different construction. Here we describe the widely spread Affymetrix^{TM} platform. The probes are made of excerpts of gene sequences, with a typical length of 25 nucleotides, and probes for one gene are spread over the chip in order to decrease the influence of systematic spatial errors. Moreover, associated to each probe is a *mismatch probe*, where one nucleotide has been replaced by its complement, thus providing the means of estimating the amount of non-specific, or false positive binding. The mismatch probe and the perfect match probe constitute a probe pair and typical arrays hold 16-20 probe pairs per gene. As opposed to spotted microarrays, high-density oligonucleotide microarrays are single-channel, measuring one sample on each array. To prepare the target, mRNA is extracted from the cell and fluorescently labeled while converted to complementary RNA. The targets are hybridized to the chip and an image is generated using a laser scanner. An image from an oligonucleotide array is slightly more standardized than a spotted microarray image in terms of location and size of the probes on the image, but basically the same image processing steps as for spotted microarrays need to be performed - localization, segmentation and intensity estimation. Let PM_{ik} and MM_{ik} be the extracted perfect match and mismatch intensities

for probe pair k of gene i, where $i = 1, ..., n$ and $k = 1, ..., K$ with K being the number of probe pairs. For each probe pair we may, as in the previous case, estimate the expression level as

$$Y_{ik} = PM_{ik} - MM_{ik} \tag{2.3}$$

Some approaches, however, disregard the mismatch intensities altogether and let $Y_{ik} = PM_{ik}$. The estimation of the expression level of a particular gene requires the summary of the probe pair expressions Y_{ik}, $k = 1, ..., K$ in one single value - an *expression index* as it is termed for oligonucleotide arrays. A straightforward way to do this is to compute the average, that is, $Y_i = \sum_k Y_{ik}/K$. However, this is sensitive to outliers, so instead a trimmed mean can be computed, which is defined as

$$Y_i = \sum_k w_{ik} Y_{ik}, \qquad w_{ik} = \begin{cases} 1/\#A, & \text{if } k \in A; \\ 0, & \text{otherwise,} \end{cases} \tag{2.4}$$

where A is the set of probe pairs such that Y_{ik} is within three standard deviations from the mean.

2.3 Gene expression data analysis

The data processing, from scanned array images to the final biological interpretation involves a long series of computational manipulations and analysis of the data, each one, in their own respect, more or less challenging. We have already, in the previous section, described the initial steps - the estimation of expression levels from the raw image data through spot identification, segmentation, intensity estimation and the computation of expression indexes. This is followed by a number of *preprocessing* steps, where various transformations of the data is applied in order to filter out non-biological variation and to 'clean up' the data to facilitate the subsequent analysis. At this stage, data is presumably ready to be analyzed in search for a biological interpretation. A range of *high-level analysis* methods exist that have as a common aim the extraction of biologically relevant patterns and information from the data. Clustering, classification, dimensionality reduction and other types of methods are frequently applied in gene expression data analysis. Finally, the extracted structure needs validation, and here too, computational methods are helpful, for example to compare the results to prior knowledge which is often stored in large databases.

2.3.1 Preprocessing

In the next sections we provide four steps for transformation of measured gene expression data: *data transformation*, *normalization*, *missing value imputation* and *filtering*, in order to prepare the data for subsequent high-level analysis.

2.3.1.1 Data transformations

As described in Section 2.2, expression values on a spotted microarray are computed as the logarithm of the ratio between the two channel expressions. The reason for taking the logarithm is to symmetrize between up- and down-regulation. In the original scale, down-regulation (that is when the query target is less abundant than the reference target) is squeezed in the interval [0, 1], while up-regulation is spread over the interval [1, ∞). Taking the logarithm makes up- and down-regulation symmetric in the interval (-∞, +∞).

2.3.1.2 Normalization

Any given set of microarray measurements contains variation originating from different sources. The expression levels may vary across samples due to differences in the quantity of mRNA, different sample processing, scanner calibration, etc. Naturally, it is of interest to remove technical and experimental variation so that what remains is the biological variation, relevant to the study. This is the objective of *normalization*.

The sources of variation are many and all are not very well understood, therefore it is difficult to model them explicitly. Instead, typically some general assumption about invariance of certain quantities over samples is made. For example, in *total-intensity normalization* it is assumed that the true average gene expression is constant across samples, in which case each array is scaled by its total estimated expression. An extension of this idea is used in *quantile normalization* (15), where the distribution of expression values is assumed to be constant across samples. In this case, estimated expression values are transformed so that in a multidimensional quantile-quantile plot of the sample distributions, the quantiles lie along the main diagonal.

Under some circumstances none of the assumptions underlying the normalization methods is valid. This is, for example, the case if the microarray contains relatively few probes, the majority of which are known to be involved in the biological process under study. In this case, normalization is often based on the assumption that expression properties of a subset of the genes are invariant. This subset can be genes that are biologically known to have a constant expression, so called *housekeeping genes*, or it can be so called *spike-in genes* from some other organism whose mRNA is added in known amounts early in the experimental process.

2.3.1.3 Missing value imputation

The spotted microarray datasets, in particular, often come with missing values. Various spots may have been flagged as unreliable, for example due to scratches or debris on the slide, and therefore lack expression values. In a study with many samples it is quite likely that a rather large fraction of the genes contain at least one missing value across samples. High-level analysis methods usually do not allow missing values therefore in order not to

throw away too much potentially valuable information, the missing values need somehow to be filled in.

Different strategies to achieve this *missing value imputation* exist. A crude approach is to use the average expression value of the gene across samples. Another solution is adopted in *K nearest neighbor imputation*, where, if gene i contains a missing value in a particular sample, the K genes with most similar gene expressions in the rest of the samples (where the corresponding sample has a value) are found and the missing value is replaced by a weighted average of the values in the other genes (40).

2.3.1.4 Filtering

Filtering is often applied to a microarray dataset prior to high-level analysis. By discarding genes that have noisy expression levels it is believed that the performance of subsequent high-level analysis increases.

Different rules are applied in order to filter genes. For example, the Affymetrix^{TM} oligonucleotide microarray platform provides *detection p-values* estimating the confidence of the signal presence of each gene and filtering can thus be based on these p-values by requiring that a gene should be significantly present in at least a certain number of samples. For spotted microarray data, one can use similar criteria based on the ratio between foreground and background intensities or the fraction of missing values.

2.3.2 High-level analysis

Once data has been properly preprocessed, the next step is to extract some biological meaning from it. A multitude of tools from the fields of statistics, pattern recognition and machine learning are helpful for this purpose. This section reviews different types of methods that are adopted to extract different types of information.

Generally speaking, the high-level analysis is a problem of mapping the expression data into some particular representation system, the choice of which will depend on the kind, and level, of structure we wish to infer. First we need to settle how to mathematically represent the expression dataset. Suppose that m measurements of the expression levels of n genes are given. Let x_{ij} be the estimated expression level of gene i in sample j and arrange the data in a matrix X where, thus, each row g_i, $i = 1, ..., n$ represents the expression levels of a particular gene, and each column x_j, $j = 1, ..., m$ represents the expression levels of a particular sample. We may think of this set of data in two ways. The first is to look at the m samples as points in an n-dimensional *gene expression space* where the coordinates of a sample are given by the expression levels of its genes. Alternatively, we can consider the n genes as points in an m-dimensional space, the *sample expression space*, where the coordinates of a gene are given by its expression levels in the different samples.

While the mathematical representation of input data as vectors in expression space is quite obvious, the choice of representation of underlying patterns is more interesting. In clustering and classification, we commonly represent structure simply by class labels. An object (gene or sample) is thus described by a single integer, determining which partition of the objects it belongs to. Another structure representation is adopted in dimensionality reduction and regression, where data is mapped into the Euclidean space, and thus each object is described by a set of coordinates in this space. The most complex structure representation is the one commonly adopted in gene network inference, where objects (in this case, genes) correspond to nodes in a graph structure, and where the structure we wish to infer is the graph edges with their respective weights. To summarize, we may write:

$$\left.\begin{array}{rl} \text{gene expression space} & \mathbb{R}^n \\ \text{sample expression space} & \mathbb{R}^m \end{array}\right\} \ni x \to z \in \left\{\begin{array}{cl} \mathbb{Z}_p & \text{clustering, classification} \\ \mathbb{R}^d & \text{dim. reduction, regression,} \\ \langle V, E \rangle & \text{gene network inference} \end{array}\right.$$

where $\langle V, E \rangle$ are vertices and edges of a weighted graph.

In parallel with the framework described above, a common way to classify different problems in data analysis in general is to discriminate between *supervised* and *unsupervised* problems. Supervised problems assume the existence and use of prior knowledge, such as classification, while unsupervised problems do not.

Microarray datasets have some specific features that have implications for what kind of information can be extracted from it. First, they typically contain many more variables (genes) than observations (samples), while classical statistics typically assumes the study of many samples described by relatively few variables, carefully chosen based on prior knowledge, to describe a particular phenomenon. For data from microarrays, as well as from several other emerging high-throughput measurement techniques, this is not the case. There is usually an abundance of variables, whereas perhaps only a few of them might be relevant. Second, due to the existence of genetic regulatory networks, there are complex dependency structures between genes, therefore the common assumption of independence between variables can not be made. These features lead to difficulties when using microarray data for rigorous tests of hypotheses on a genome-wide level. However, testing hypotheses is not the only use one can have of data. *Generating* hypotheses is often equally valuable and it is important that we also cover this approach in the gene expression data analysis.

The rest of this section describes differential expression analysis, classification and clustering in gene expression data analysis. Please note that methods for classification and clustering of gene expression data are not performed in this work, but we present them here in order to introduce the reader with the most important high-level analysis methods.

2.3.2.1 Differential expression analysis

One of the most immediate questions in a study of a microarray dataset is which genes are differentially expressed (DE) in two or more specified groups of samples. This problem is studied in *differential expression analysis*. Answers typically come in the form of gene lists which can be further studied in the search for biological insights to, for example, disease mechanisms.

For simplicity, suppose that we want to find DE genes in a two-group comparison. The most common approach is to study gene by gene and select those that show differential expression in the two groups. An early methodology for this is *fold analysis*, where the amount of differential expression is measured by the expression ratio between the two samples and differential expression is considered significant if it is above or below constant threshold values, typically 2 and 0.5, respectively (70).

Current datasets usually contain several samples per group. In this case, the use of statistical tests like some *Student's t-test* become possible. For each gene, we may, for example, compute the two-sample t-statistic

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1}{N_1} + \frac{\sigma_2}{N_2}}} \tag{2.5}$$

where μ_1, μ_2 are group means; σ_1, σ_2, estimated standard deviations and N_1, N_2, number of samples in group 1 and 2, respectively. The distribution of the statistic under the null hypothesis, which is that the gene is not differentially expressed in the two groups, can either be assumed to follow a t-distribution or be estimated by permutation of the class labels. In this way, a *p-value*, quantifying the significance of the differential expression, can be obtained.

Microarray datasets involve many more variables (genes) than observations (samples) and this needs to be taken into consideration while looking for DE genes. For example, consider a dataset where the number of genes is $n = 10^4$ and the fraction of truly non-DE genes is $\pi_0 = 0.9$. Suppose $p = 0.05$ is chosen as a threshold for calling a gene differentially expressed. The expected number of non-DE genes that is incorrectly called differentially expressed is then $p \cdot n \cdot \pi_0 = 450$ - almost half the number of truly DE genes. Moreover, not all DE genes will be called DE since, in some cases, by chance, the p-value will exceed 0.05. The result is that the gene list will contain a large fraction of genes that have nothing to do with the condition that defines the groups. This exemplifies the *multiple testing problem*, where a p-value threshold that seems standard and reasonable in a single test leads to a high *false discovery rate (FDR)*, defined as the expected proportion of false positives among the declared significant results. Discussions and procedures for minimizing the FDR are given in Chapter 3.

The t-test has also several cons for testing the differential expression of genes. For example in the presence of outliers, the expression distributions are non-normal, and we can

not use t-distribution for inferring the p-values. Another problem is that a small number of samples makes it difficult to get significant results in the tests. A third problem is that gene expressions are typically not independent, so the assumption of independence is violated. However, if genes are, on average, uncorrelated and the dependencies are weak, it can be argued that this problem is less severe.

Treating genes one-by-one does not make use of the full potential in a microarray dataset. Since groups of genes are correlated it makes sense to perform differential expression analysis of whole groups of genes instead of single genes. Such groups may, for example, be defined as genes known to be involved in particular pathways of interest or as genes located in the same chromosomal regions. Gene set enrichment analysis (GSEA) (75) implements a method for this, and it has been shown that, using this method, differential expression of groups of genes can be identified where single gene tests would fail to discover differential expression. GSEA and other methods for group differential expression, also called *group enrichment analysis*, are presented in Chapter 3.

2.3.2.2 Classification

The task of *classification* is that of learning how to best guess which, out of a number of given classes, an object with unknown class label belongs to. A classifier is constructed by training, where it is introduced to representative training objects of known classes - the *training* set. For microarray data, the applications of classification include diagnosing cancer type, given the expression pattern from a tumor sample, or predicting the biological function of genes based on their expression patterns.

In a sense, classification is similar to differential expression analysis, since most classification algorithms, explicitly or implicitly, work by finding variables, or functions of variables, that are good predictors of the class. A difference, however, is that while a biological interpretation of these predictors is a nice side-effect, it is not the primary goal as in differential expression analysis. The similarity is particularly clear in a class of methods represented by (32), where, given a two-group classification problem, a list of differentially expressed genes is extracted using the training set, and a voting function, defined on these genes, decides which class a presented sample belongs to. A problem with this approach is that many of the discriminative genes are likely to be correlated, perhaps being involved in the same particular process. Genes with less strong differential expression, but uncorrelated with the group of most strongly DE genes, would most likely increase the generalization performance but are not included using the basic method described.

In gene list based classifiers, as above, the decision function is fixed and predefined while the set of variables on which it operates is learned from the data. Other methods, such as *Artificial Neural Networks (ANNs)* and *Support Vector Machines (SVMs)*, use all variables but let the method learn the decision function from the data. Here too, a list of predictive variables can be extracted by ranking them according to their influence on

the final decision function. Artificial neural networks can readily be applied to classification problems with more than two classes (44), while support vector machines are binary classifiers, discriminating between, for example, healthy and cancerous tissue (29), or classifying genes as belonging to a known functional group or not (17). Binary classifiers can be extended to handle K classes by learning to classify between each pair of classes or by learning K classifiers of one class against all others. Both ANNs and SVMs are able to learn nonlinear decision functions.

Just like in differential expression analysis, the fact that the genes by far outnumber the samples, introduces some difficulties. For example, gene list based classifiers, as described above and in (32), classify judging from lists of top discriminatory genes. As pointed out in the discussion on differential expression analysis, such lists are likely to be 'infected' by false positives, which by their presence disturb the classification of new samples. In fact, this problem is common to all classification methods when the number of samples is much smaller than the number of variables. A classifier risks to trust variables or sets of variables as having predictive power when in fact they have this power (on training data) by chance.

A wealth of different classification methods have been applied to microarray data but, so far, none has stood out as significantly more suitable than the others. Presumably, the importance of the method choice will grow with the number of samples in the datasets.

2.3.2.3 Clustering

Clustering is the process of grouping together similar objects into resulting groups, or clusters. In gene expression data analysis, clustering serves to discover groups of co-regulated genes or groups of samples with similar expression profiles, for example revealing classes or subclasses of disease states. The problem is similar to that of classification, with the difference that clustering methods discover groups in data without using any prior knowledge, while classification methods do so by arranging objects into class labeled groups. Indeed, classification methods are sometimes referred to as *supervised clustering*.

The problem of grouping together 'similar' objects calls for a definition of similarity. There are several ways for calculating the (dis)similarity of two vectors, but two most frequently used measures are Euclidean distance and correlation distance, defined as $1-\rho_{xy}$, where ρ_{xy} is the correlation between vectors x and y.

Hierarchical clustering. *Hierarchical clustering* is currently the most frequently used clustering method in gene expression data analysis,(27) being an early example. The (agglomerative) hierarchical clustering algorithm takes as input a matrix of pairwise similarities between objects. Initially all objects are considered as clusters. Then, iteratively, the most similar cluster pair is found and merged together into a new cluster. This is repeated until all objects are contained in a single cluster. Similarities between two clusters can be defined in a number of ways, for example, the largest similarity between any pair of objects in separate clusters (*single linkage*), the smallest similarity between any pair of objects in

separate clusters (*complete linkage*) or the average similarity between all pairs of objects in separate clusters (*average linkage*). The clustering is visualized in a cluster tree, a *dendrogram*, visualizing the nested structure of clusters. Hence, in fact, hierarchical clustering yields a more detailed structure representation (a tree-graph) than many other clustering methods that simply divide the data into partitions.

Hierarchical clustering is frequently used in comparative genomics and phylogeny to study, for example, the evolutionary development of gene sequences, and perhaps hierarchical clustering is more suited for data where distances can be defined as a discrete number of alterations, than for quantitative data like gene expression data. One problem is that it might be difficult to decide which clustering level in the dendrogram to choose, if the aim actually is to partition the data. On the other hand, the user does not have to provide an a priori number of clusters. Another issue is over-fitting. Different ways of defining the similarities between points and clusters yield very different cluster trees. Hence, there is a risk of adjusting parameters until getting a tree that adheres to prior beliefs.

K-means clustering. *K-means clustering* is a standard and well understood clustering algorithm. The algorithm takes as input the expression data and the number of clusters, K. Initially, K cluster centers are randomly placed in the span of the data. All objects are assigned to their nearest cluster center and the mean expression of each cluster is calculated. These means replace the prior cluster centers and the two steps are repeated until convergence. The advantage of K-means is that the method does not have many parameters to assign, while in many cases it is a drawback that the number of clusters has to be provided to the algorithm. An example of the use of K-means clustering in gene expression data analysis can be found in (77).

It is important to keep in mind that most clustering algorithms will divide data into clusters even if no real cluster structure is present. Therefore, it is important to control the significance of the produced results. For example, when clustering genes, measured over a small number of samples, the clusters will contain many false positives, while many true positives will be missed. This is due to the fact that the statistical confidence of similarity measures will be low.

2.3.3 Further analysis

Once high-level analysis methods have suggested some underlying structure in the data, these results need to be interpreted and validated in terms of biological significance. This can be done in a number of different ways. Suppose, for example, that we have clustered the data, so that what we have to validate is a particular partition of the data.

A natural way to validate sample clusters is to consult clinical variables of the samples (e.g., gender, blood pressure, cancer diagnosis) and investigate if patterns, similar to the ones discovered in the gene expression data, appear there. With an extensive clinical

database this might be a task at almost the same complexity level as the gene expression analysis itself. One particular way of evaluating proposed *sample clusters* in diseases like cancer is the use of *Kaplan-Meier survival analysis*, which involves a statistical test of whether two groups of patients have significantly different median survival times. If this is found, it is often argued that the clusters are of biological relevance, for example, as subgroups of the same disease (5).

For genes, the available knowledge does not come in the shape of sets of clinical variables like for samples. Instead, *gene clusters* can be validated with respect to, e.g., cellular functions, chromosomal locations, sequence information, etc. Knowledge about the genome is stored in Gene Ontology (GO) (6) databases where genes are arranged in tree structures according to function, location and other properties. Given a set of genes, one can make a query to a GO database testing if some, say functional, group on some tree level is over-represented among the genes (91). Alternatively, databases containing known pathway relations, such as the Kyoto Encyclopedia of Genes and Genomes (KEGG) (41), may be consulted to see whether genes from some particular pathway are over-represented in the cluster. The evaluation of the meaning and relevance of gene clusters using queries to databases requires that enough useful information has been stored in the database by human curators. To circumvent this, one may make use of text mining techniques which search through vast collections of literature and attempt to extract relevant information. For a given gene cluster, text mining methods can retrieve abstracts where subsets of the given genes co-occur. From these abstracts, overrepresented keywords can be extracted in order to gain understanding of the functional or clinical context of the genes (61). On a more detailed level, the literature can be mined for causal relations between the genes, thereby suggesting a network of functional relations between genes (39). A third way to evaluate gene clusters is to consult sequence data. The upstream regions of the genes are then searched for shared subsequences, presumably corresponding to known or unknown transcription factors. The existence of such shared subsequences may then confirm the biological relevance and aid the understanding of the role of the gene cluster.

In this thesis we focus on extracting biological knowledge stored in popular bio-ontologies, as GO and KEGG, so in the next section we give a brief overview of the controlled biological vocabularies maintained by the Gene Ontology Consortium, followed by an introduction to biological pathway database KEGG and related gene annotations.

2.4 Gene Ontologies

Recent developments in molecular biology have brought an explosion in the amount of available data. The sequencing race began in 1996 with the release of the genome of *Saccharomyces cerevisiae*, a higher model organism commonly known as baker's yeast. The completely sequenced human genome was announced in 2003, containing approximately

3 billion base pairs and 25,000 genes[1]. Today, numerous other animal genomes are fully available, while others are still at different stages of completeness.

Information gained from completed sequences of various genomes suggests that there exists a single finite superset of genes and proteins, most of which are conserved in many or all living cells. This recognition has lead to the unification of biology. Known properties and functions of genes and proteins in a specific genome contribute to the general knowledge base, as it is likely that similar functions are expressed in homologous[2] genes of many other diverse organisms (6). Relevant information can be extracted from previously proven results with well-known model organisms, and used for studying more complex genomes.

Unfortunately, the pace at which new genomes are sequenced often exceeds the speed of organizing and cross-referencing existing data. Biological information concerning genes, proteins and their functions is primarily available in numerous genome-specific databases and maintained by different organizations. Moreover, there is often no clear understanding or agreement concerning common genetic terminology and functional descriptions of biological objects. Therefore, the task of finding relevant genes of similar function may be quite challenging.

The Gene Ontology (GO) Consortium was established to address the above problems. The primary goal of the consortium is to maintain and develop a controlled and organism-independent vocabulary of the molecular biology domain. Such a vocabulary provides a hierarchical collection of terms, to describe general as well as specific molecular functions, cellular components and biological processes (6). The GO project was initially a collaboration between the Saccharomyces Genome Database of baker's yeast, Mouse Genome Informatics of common house mouse, and Flybase, the database of fruitfly. More databases joined the consortium later, and many other data sources are using the ontologies today for identifying genes and proteins by their functionality. Vocabularies and gene annotations are freely available at the GO Consortium web site[3].

2.4.1 Ontology design and implementation

The concept of *ontologies* in computer science was first introduced in research related to Artificial Intelligence and Knowledge-Based Systems with the purpose of sharing knowledge and improving communication between independent systems. In this terminology the body of formally represented knowledge is based on conceptualization: *objects*, *concepts* and other *entities* that are assumed to exist in some area of interest, and *relationships* that hold among them. The specific area of interest is often referred to as *domain*, and objects are known as *terms*.

[1] National Human Genome Research Institute, http://www.genome.gov
[2] Similar in position, structure, function, or characteristics
[3] http://www.geneontology.org

Ontology is an explicit specification of conceptualization. An ontology consists of a set of terms represented in a given domain, and relationships that hold between terms. Knowledge concerning objects and relations is stored in a representational vocabulary. In addition to objects and relations, ontology holds respective human-readable descriptions, and formal axioms that constrain the interpretation and the use of objects and relations.

Gene Ontology vocabularies are structured in a form of *Directed Acyclic Graphs (DAG)*, directed graphs with no path starting and ending at the same vertex. A vertex of the GO graph corresponds to a biological term, and a directed edge between two terms shows that one term is hierarchically related to the other. Such a graph represents a hierarchical structure resembling a tree, except that each child vertex may have more than one parent vertices. The situation that a specific term is a child of multiple broad terms, captures well the biological reality.

Two types of parent-child relationships are defined in GO. Relation type *is_a* describes the fact that a child term is an instance of the parent, while relation type *part_of* denotes that a child term is a component of the parent. A child term may have different classes of relationships with its parents. Every term has a unique identifier (e.g., GO:0000001) and a name. Besides these, a number of optional properties may be defined.

There are a few GO rules and guidelines to be followed. *True Path Rule* is the most relevant guideline in the context of this work. It states that for any given child term, the path to its top-level parent must always be true. In case of multiple parents, all paths from a term to the top hierarchy have to be verified. An example GO hierarchy is shown in Figure 2.3.

2.4.2 Three ontologies of GO

As stated in the above definitions, an ontology represents knowledge of a specific domain or an area of knowledge. Gene Ontology maintains vocabularies of three domains, *Molecular Function*, *Biological Process* and *Cellular Component*. These particular classifications were chosen because they represent information sets that are common to all living organisms. Vocabularies are developed for a generic eukaryotic cell (cell that have nucleus); specialized organs and body parts are not represented.

It is correct to say that Gene Ontology consists of three independent vocabularies of different domains. Each vocabulary has one root term and there are no parent-child relations linking vertices of different ontologies.

- *Molecular function* (GO:0008639, *MF* or *Func*) is defined as what a gene product does at the biochemical level. Domain terms only specify function, location and time of event remain undefined within ontology.

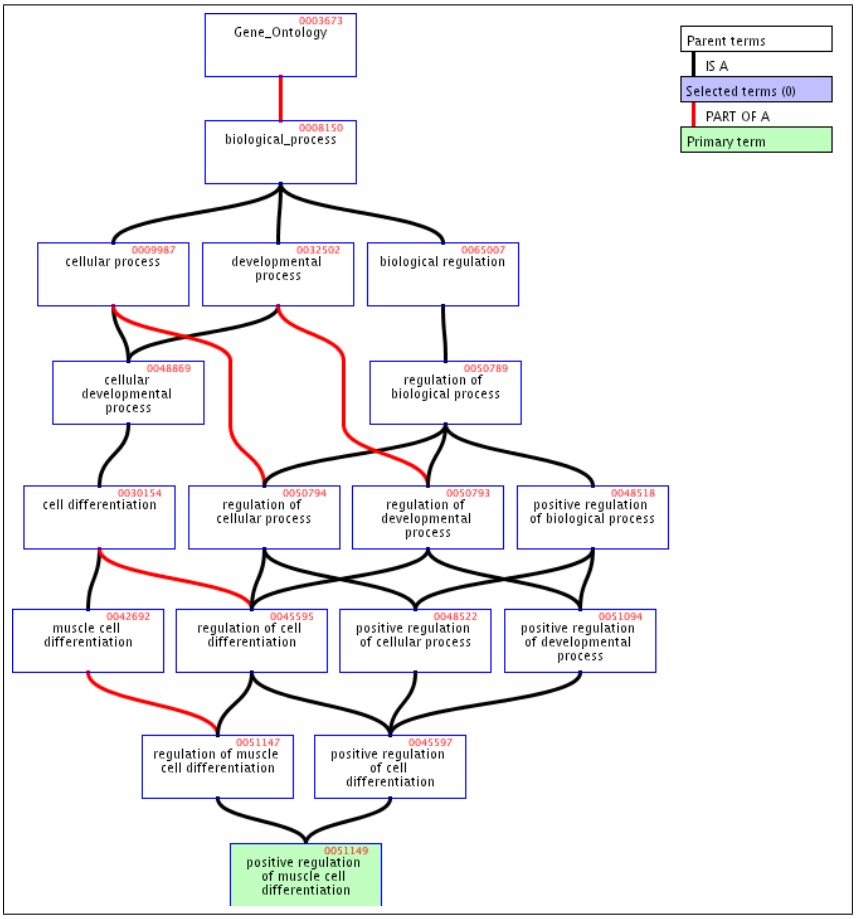


Figure 2.3: A part of the GO providing the annotations concerning the positive regulation of muscle cell differentiation.

- *Biological process* (GO:0008150, *BP* or *Proc*) refers to the biological objective to which a gene product contributes. A process is accomplished by one or more ordered assemblies of functions, often involving transformation of biological matter.
- *Cellular component* (GO:0005575, *CC* or *Comp*) refers to the place in a cell or extracellular region where a gene product is found or where the product is active.

Two types of terms deserve further attention. Every domain has an *unknown* term just below the root; it is meant to hold genes and gene products that have been investigated, but no knowledge of the domain has been revealed. From time to time, some terms are marked as *obsolete* as biological knowledge evolves; these terms are removed from active vocabularies and placed under *obsolete* term of the domain in question.

The GO Consortium explicitly states that Biological Process domain is not equivalent to a biological pathway and describing a pathway through the necessary dynamics and dependencies between processes and functions is beyond the scope of the GO project.

The GO Consortium recognizes that there exists a biological relationship between a series of molecular functions in a biological process, that unfold in a certain component of a cell. This means that there are in fact numerous interconnections between three independent domains. Even though GO could be logically expanded to reflect states, operations and components of cells, the current goal of the project is to concentrate on the development of three independent and precise collections of terms.

As of September 2007, GO vocabularies consisted of 21,908 terms, including 12,549 terms of Biological Process, 1,846 terms of Cellular Component and 7,513 terms of Molecular Function. There were 1001 obsolete terms not included in the above statistics. The longest path from child to root involves 15 edges, but most of the terms are normally distributed at middle levels (Figure 2.4). Ontologies are by no means complete and are continuously expanding through collaboration of many organism-specific databases.

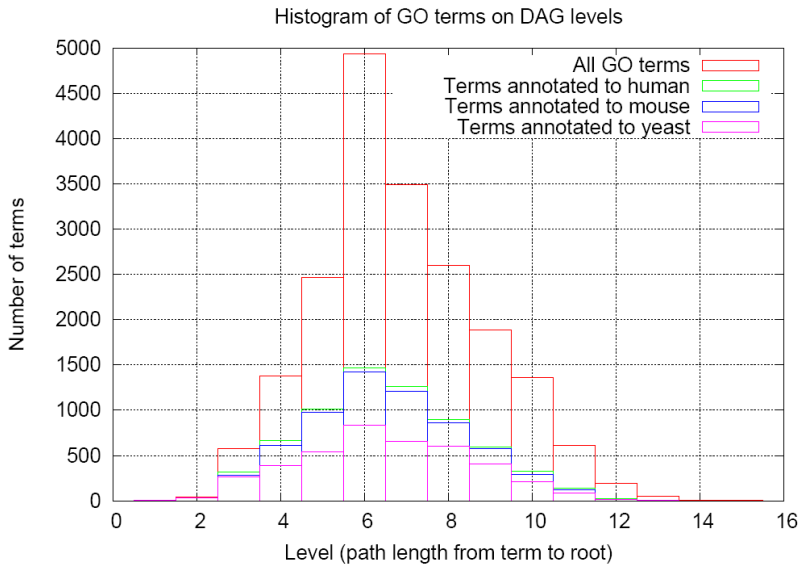


Figure 2.4: Histogram of the GO terms on different DAG levels.

2.4.3 Gene annotations

In addition to the three GO ontologies of Molecular Function, Biological Process and Cellular Component, several databases exists that include links between GO terms and genes or gene products. These links are commonly referred to as *annotations* or sometimes as *associations*.

The GO Consortium maintains annotations of three model organisms of founding members, namely baker's yeast *Saccharomyces cerevisiae*, common house mouse *Mus musculus*

and fruitfly *Drosophila melanogaster*. GO annotations of numerous other genomes, including human, are now available at National Institute of Health runned web site ENTEZ[1].

Every annotation to GO is attributed to a source, which may be literature reference, another database or computational analysis. Annotation must also indicate the type of evidence, provided by the source to support the association between the given entity and the GO term. A standard set of *evidence codes* is available for qualifying annotations with respect to different types of experimental conditions (7). Evidence codes provide means to describe a range of different experiments varying from *in vitro*[2] techniques to purely *in silico*[3] methods. The GO web site includes a comprehensive annotation guide for evidence codes and proposes a loose order of decreasing reliability.

One of the most important guidelines of GO is the previously described True Path Rule. In the context of annotations, the guideline is interpreted as follows. Every gene or gene product that is annotated to a specific term in the GO, is always annotated to all term's parents up to the top-level parent, using all possible paths from the term to the root (6). Such indirect True Path annotations are not provided in GO datasets and therefore need to be inferred explicitly. The True Path Rule also explains the need for storing several evidence codes for any gene-term pair. Besides the fact that different experimental results may support exactly the same annotation, terms located in the top of the hierarchy get repeated indirect annotations of same genes via different paths.

Figure 2.5 displays a histogram for term sizes in the sense of the number of annotated genes. For every organism, there are numerous highly specialized terms with only a few annotated genes, while larger groups are more uncommon. Largest groups on the right side of the figure represent root terms, each of these containing the union of its descendants' annotations.

In this section, we give a brief introduction to biological pathways and then in Chapter 4 we propose a simple model for integrating knowledge from Gene Ontology vocabularies with pathway databases, such as KEGG, and gene-gene interaction data provided in the ENTEZ database.

2.4.4 Biological pathways

According to (43), a *pathway* is a linked set of biochemical reactions, where a product of one reaction is a reactant of, or an enzyme that catalyses, a subsequent reaction. In other words, a pathway is a biochemical process that can be partitioned into component steps. Small metabolic processes with just a few reactants, as well as macroprocesses involving

[1] http://www.ncbi.nlm.nih.gov/entrez/, for downloading structured gene information provided on the web site visit ftp://ftp.ncbi.nlm.nih.gov/gene/

[2] Latin: "within glass"; biological experiments performed in a test tube, or generally outside a living organism or cell.

[3] Latin: "within silicon"; a general term for any computational means in biology.

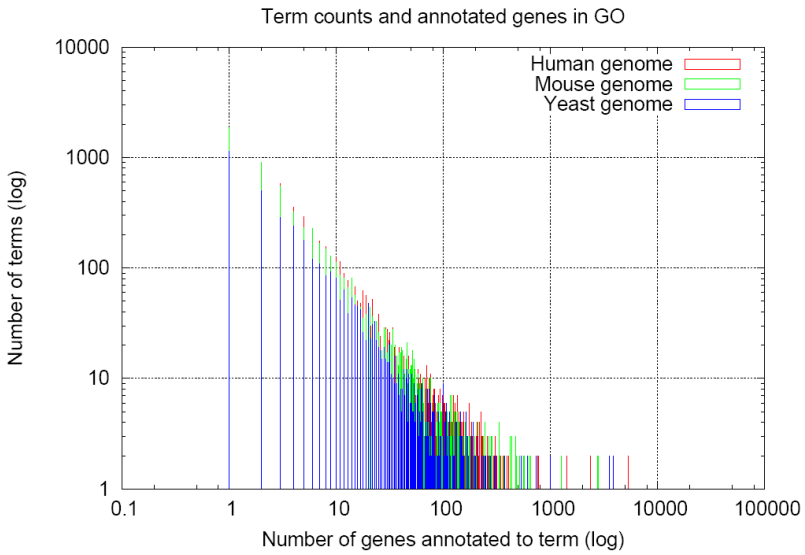


Figure 2.5: Histogram for sizes of GO term annotations.

hundreds of molecular components with the cooperation of multiple cells, are commonly described as pathways (43).

Metabolic networks are currently the most well-studied biological pathways. A *metabolic network* is essentially a chemical processing factory within each cell, that enables the organism to convert small molecules from the environment into building blocks of its own structures, and to extract energy from these molecules.

A sequence of steps comprising a pathway is rarely a simple linear sequence, as a single reaction often requires multiple inputs and creates multiple outputs. A pathway may contain redundancy, as multiple parallel series of events produce the same biochemical result. On the other hand, a single molecular component can be multifunctional and involved in multiple pathways with different goals. Pathways may also be competitive; activities of one pathway may render the other pathway inactive, as the first one consumes, binds or deactivates some resource on which the second pathway depends (69).

A mathematical representation for a pathway is a directed graph, that at a high level displays the cause-effect dependencies among the components. It has been more common to display molecular components as nodes of a graph and underlying events (reaction, modification, translocation, transcription) as edges between the nodes (69). Figure 2.6 graphically displays the *glycolysis and gluconeogenesis* pathway.

In our work we study pathway data from the The Kyoto Encyclopedia of Genes and Genomes (KEGG). KEGG is a knowledge base for systematic analysis of gene functions in terms of networks of genes and molecules, that provides means of linking genomes to

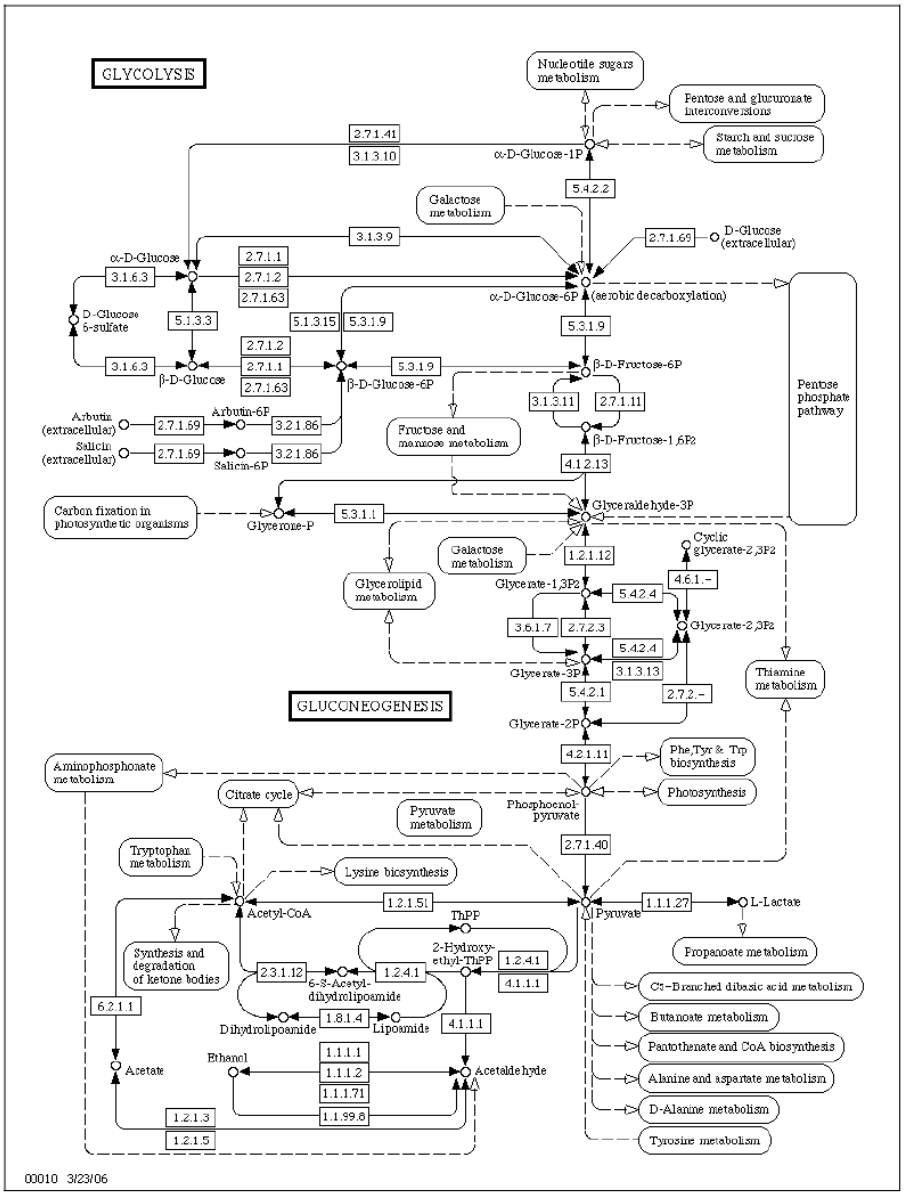


Figure 2.6: KEGG pathway 00010 for *glycolysis and gluconeogenesis*. The pathway involves 48 genes in yeast, 55 genes in mouse, and 63 human genes.

biological systems. KEGG database is publicly available on their web site[1].

KEGG resource consists of 4 major components. GENES database is a collection of gene catalogues for all complete genomes and some partial genomes. LIGAND database describes building blocks of the biochemical space, such as enzymes, chemical compound structures, reactions and other substances in living cells, as well as a set of drug molecules. PATHWAY database consists of a collection of pathway maps, while BRITE database

[1]http://www.genome.ad.jp/kegg/pathway.html

holds a collection of hierarchies and binary relations that correspond to rules governing the genome-environment interactions in pathways.

In current work, we are most interested in the KEGG PATHWAY database, that holds a collection of manually drawn pathway maps for metabolism, genetic information processing, environmental information processing, and other cellular processes (41).

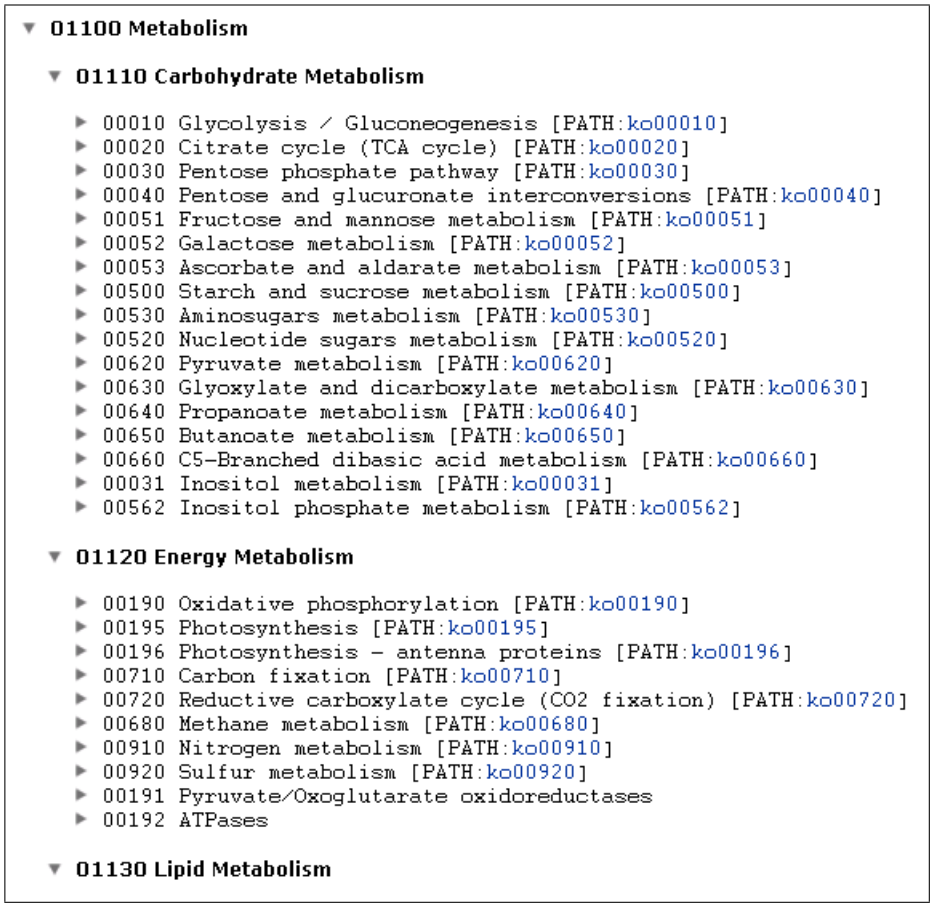


Figure 2.7: This figure shows a part of the KEGG Orthology providing the annotations concerning Carbohydrate and Energy Metabolism.

Every pathway in KEGG is identified with a five-digit code (00010) and described with a name (glycolysis and gluconeogenesis). Organism-specific pathways are automatically generated based on the generic pathway maps by matching genes from the organism's catalogues. Pathways are partially distributed into classes and subclasses. For example, the broad class metabolism is divided into subclasses like energy metabolism, nucleotide metabolism, etc. Each of the subclasses holds a number of pathways, for example sulfur metabolism is a kind of energy metabolism. This pathway organization and structuring is called the *KEGG Orthology* (KO) (see Figure 2.7).

In the next chapter we present three basic methods for calculating gene set enrichment, that are used in functional interpretation of gene expression data: Fisher's exact test, Gene Set Enrichment Analysis (GSEA) and Parametric Analysis of Gene set Enrichment (PAGE). Together with these methods, we also present the problem of Multiple Testing, and some techniques for its solution.

3 Functional Interpretation of Gene Expression Data

Molecular biology has addressed functional questions by studying individual genes, either independently or a few at a time. Despite its reductionistic approach, it was extremely successful in assigning functional properties and biological roles to genes and gene products. The recent possibility of obtaining information on thousands of genes or proteins in a single experiment, thanks to high-throughput methodologies such as gene expression or proteomics, has opened up new possibilities in studying living systems at the genome level that are beyond the old paradigm 'one-gene-one-postdoc'. Relevant biological questions regarding genes, gene products interactions or biological processes played by networks of components, etc., can now for the first time be addressed realistically and used in the more advanced analysis of biological results.

Nevertheless, genomic technologies are at the same time generating new challenges for data analysis and demand a drastic change in data management. Dealing with this abundance of data must be approached cautiously, because of the high occurrence of spurious associations, if the proper methodologies are not used and if statistical testing is not applied rigorously.

To translate this abundance of data into information, numerical analysis is firstly required to determine which genes (among the thousands analyzed) can be considered as significantly related to the phenotypes (see Section 2.3.2.1). The second step is to interpret the roles played by the targeted genes. The availability of GO and KEGG annotations for a considerable number of genes helps interpret these results from a biological point of view.

The hypothesis commonly used is as follows:

> *if some genes have been found to be differentially expressed when comparing two different phenotypes (or are correlated to a given continuous phenotypic trait, or to survival, etc.) it is because the roles they play at the molecular level account (to some extent) for the phenotypes analyzed.* (3.1)

The GO and KEGG annotations available for the genes can serve as a more or less detailed description of these biological roles. For example, if 50 genes from an array of 6,500 genes are differentially expressed and 40 of them (80% - a high proportion) are annotated as 'response to external stimulus' (GO:0009605), it is intuitive to conclude that this process must be related to the phenotypes studied. In addition, if the background distribution of this type of gene in the genome is, say 4%, one can conclude that most of

the genes related to 'external stimulus' have been altered in their expression levels in the experiment.

Using (3.1) as a basis for functional interpretation of gene expression data, in this chapter we present three methods that improve the analysis. The first one is from the class of threshold-based interpretation, where first genes of interest are selected, and then their annotation is analyzed, using the biological background knowledge provided in the annotation databases. The last two are from the class of threshold-free interpretation, where first genes are ranked by using their differential expression values (e.g., using the t-scores), and then positions of members of predefined gene sets (using GO, KEGG) in the ranked list are analyzed using appropriate statistical tests (e.g., Kolmogorov-Smirnov).

3.1 Threshold-based functional interpretation

The final aim of a typical microarray experiment is to find a molecular explanation for a given macroscopic observation (e.g., which pathways are affected by the loss of glucose in a cell, what biological processes differentiate a healthy control from a diseased case).

In the first generation of approaches proposed, the interpretation of microarray data is usually performed in two steps: in the first step genes of interest are selected (see Section 2.3.2.1), because they co-express in a cluster or they are significantly over- or under-expressed when two classes of experiments are compared. The selection process does not take into account the fact that these genes are acting cooperatively in the cell and consequently their behavior must be coupled to some extent. In this selection process, under the unrealistic simplification of independence among gene behaviors, rigorous thresholds are usually imposed to reduce the false positives rate in the results. In the second step, the selected genes of interest are compared with a background (typically the rest of the genes) in order to find enrichment in any functional term. This comparison with the background is required, otherwise the significance of a proportion (even if high) cannot be determined. This comparison to the background is essential because sometimes apparently high enrichment in a given functional term is nothing but a reflection of a high proportion of this particular term in the whole genome and, consequently, has nothing to do with the set of genes of interest. The procedure for the interpretation of genes selected by significant differential expression between two pre-defined classes of experiments is illustrated in Figure 3.1.

This second step of comparison between the selected genes and the background can be carried out by means of the application of other, equivalent tests such as the hypergeometric, binomial, Fisher's exact test, etc., implemented in different available tools, reviewed in (45). Among these tools, the most popular ones (most quoted in the literature) are Onto-express (46) and FatiGO (1). These tools use different biological terms with functional meaning such as GO, KEGG pathways and other terms of biological relevance.

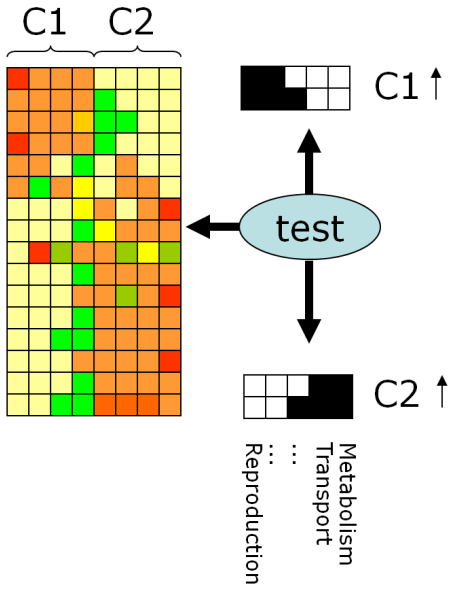


Figure 3.1: The two-step procedure for the functional interpretation of distinct microarray experiments, implementing the supervised approach with functional annotations of genes differentially expressed among two classes (C1 and C2) of experiments. The figure represents a list of genes (rows) ordered by differential expression when classes C1 and C2 are compared. Genes on the top are more expressed in class C1 (red color) than in C2. Conversely, genes on the bottom are more expressed in class C2. There is a gradient of differential expression between the two extreme situations. Typically genes are arranged by means of a test (e.g., the *t*-test) and those with a value of the statistic over a given threshold are declared as significant (right part). Then the distribution of functional terms (e.g., GO terms) among the differentially expressed genes and the rest is compared by means of another test (e.g., the Fisher's exact test).

Here we present the most used threshold-based procedure for calculating the enrichment of a gene sets, i.e., the Fisher's exact test.

3.1.1 Fisher's exact test

When using Fisher's test, the score for a gene set annotated by GO term S is the degree of independence between the two properties:

$$\text{A} = \text{gene is in the list of differentially expressed genes}$$
$$\text{B} = \text{gene is annotated by GO term } S$$

Testing the independence of these two properties corresponds to Fisher's exact test (57), and is computed by the following procedure:

1. Let N be the number of genes on a microarray
2. S is a GO term
   (a) M genes $\in S$
   (b) $N - M$ genes $\notin S$
3. Let k be the number of differentially expressed genes
4. The probability of having exactly x, out of k DE genes, annotated by S is computed as follows:

$$p(X = x|N, M, k) = \frac{\binom{M}{x}\binom{N-M}{k-x}}{\binom{N}{k}}$$

5. The Fisher's score determines the probability of having at least x genes, out of k differentially expressed genes, annotated by S:

$$p = 1 - \sum_{i=0}^{x-1} \frac{\binom{M}{i}\binom{N-M}{k-i}}{\binom{N}{k}}$$

3.1.2 Statistical approaches to test significant biological differences

As previously mentioned much caution should be made when dealing with a large set of data because of the high occurrence of spurious associations (30). Table 3.1 has been constructed using ten random datasets obtained by the random sampling of 50 genes from the complete genome of *Saccharomyces cerevisiae* (yeast used in baking and brewing). For each random set, the proportions of all the GO terms (at GO level 4) have been compared between both partitions (50 genes with respect to the remaining ones), and the GO term showing the most extreme differential distribution was displayed in each case (rows of the table). The first column shows the percentage of genes annotated with the GO term in the random partition of 50 genes, the second column represents the corresponding percentage in the rest of the genome and the third column shows the p-value obtained upon the application of a Fisher's exact test. It is astonishing that most of the random partitions present asymmetrical distributions of GO terms with significant individual p-values (column 3).

This apparent paradox stems from the fact that we are not conducting a single test in each partition, but as many tests as GO terms are being checked (several thousands). Therefore, the common mistake is made when a researcher tends to forget about the many hypotheses rejected and only focuses on the term for which an apparent asymmetrical

Table 3.1: GO terms found to be differentially distributed when comparing ten independent random partitions of 50 genes sampled from the complete genome of yeast.

% in random set	% in whole genome	*p*-value	adjusted *p*-value	GO term
8.33	1.86	0.0752	1	ion homeostasis (GO:0050801)
10.00	31.34	0.0096	0.6735	nucleobase, nucleoside, nucleotide & nucleic acid metab. (GO:0006139)
3.33	0.24	0.075	1	One-carbon compound metab. (GO:0006730)
4.04	8.00	0.0177	0.6599	energy pathways (GO:0006091)
3.45	0.22	0.0669	1	metabolic compound salvage (GO:0043094)
5.88	0.67	0.024	1	vesicle fusion (GO:0006906)
6.45	1.60	0.09	1	negative regulation of gene expression, epigenetic (GO:0045814)
13.79	3.97	0.028	1	response to external stimulus (GO:0009605)
16.13	4.23	0.0097	1	response to endogenous stim. (GO:0009719)
2.70	0.13	0.054	1	host-pathogen interaction (GO:0030383)

distribution was found. In some cases this situation is caused by the way in which some of the above mentioned programs work. To some extent the fact that many tests are really being conducted is hidden to the user and the result is presented as if it were the case of a unique test. If we conduct several thousands of tests simultaneously, the probability of finding an apparently asymmetrical distribution for a given GO term increases enormously. A very simple example can be used here to illustrate this concept: let us imagine to flip a coin 10 times and get 10 heads. One would certainly suspect that something was wrong with the coin. If the same operation was repeated with 10,000 different coins one or even several occurrences of 10 heads would not be considered surprising. We intuitively accept this because of the probability of having an unexpected result just by chance is high. If we were interested in checking whether an observation is significantly different from what we could expect simply by chance in a multiple testing situation then the proper correction must be applied. The fourth column of Table 3.1 shows an adjusted *p*-value using one of the most popular multiple-testing corrections, the False Discovery Rate (FDR) (14), and it is obvious that none of the situations depicted in columns 1 and 2 can be attributed to anything else than random occurrence.

Table 3.1 shows how random partitions, for which no functional enrichment should be expected, yield apparent enrichments in GO terms because the most asymmetrically distributed GO term among several thousands are chosen a posteriori. These values occur simply by chance and cannot be considered as either biologically authentic or statistically

significant. This clearly shows that multiple testing adjustment must be used if several hypotheses are simultaneously tested.

Multiple testing has been addressed in different ways depending on particular cases and the number of simultaneous hypotheses tested. Thus, corrections such as Bonferroni or Sidak are very simple to be applied but are too conservative if the number of simultaneous tests is high (86). Another family of methods that allow less conservative adjustments are the Family Wise Error Rate (FWER), that controls the probability that one or more of the rejected hypotheses (GO terms whose differences cannot be attributed to chance) is true (that is, a false positive). The minP step-down method (86), a permutation-based algorithm, provides a strong control of the FWER. Approaches that control the FWER can be used in this context although they are dependent on the number of hypotheses tested and tend to be too conservative for a high number of simultaneous tests. In this case, it would be more appropriate to control the proportion of errors among the identified GO terms whose differences among groups of genes cannot be attributed to chance instead. The expectation of this proportion is the False Discovery Rate (FDR). Different procedures offer strong control of the FDR under independence and some specific types of positive dependence of the tests statistics (13), or under arbitrary dependency of test statistics (14).

Next, we present the most used procedures for correcting the calculated p-values when we use multiple hypothesis testing, Bonferroni and FDR.

3.1.3 Multiple testing

Say that we want to perform a statistical test with a 0.05 threshold, a threshold of 0.05 means we are 95% sure that the result is significant, but we repeat the test for twenty different hypotheses. What is the chance that at least one of the tests will receive a p-value less than 0.05 ?

- Pr(making a mistake) = 0.05
- Pr(not making a mistake) = 0.95
- Pr(not making any mistake) = 0.95^{20} = 0.358
- Pr(making at least one mistake) = 1 - 0.358 = 0.642

Consequently, there is a 64.2% chance of making at least one mistake.

For example we can take the problem of selecting differentially expressed genes. If we apply the standard procedure of calculating the t-scores of the genes, from which we calculate the appropriate p-value, and we put a threshold of 0.05, applied on Golub data (32), we will find that 1045 genes are differentially expressed (Figure 3.2). Because we apply the t-test 7,074 times, we can expect that around 350 of these 1,045 selected genes are false positives.

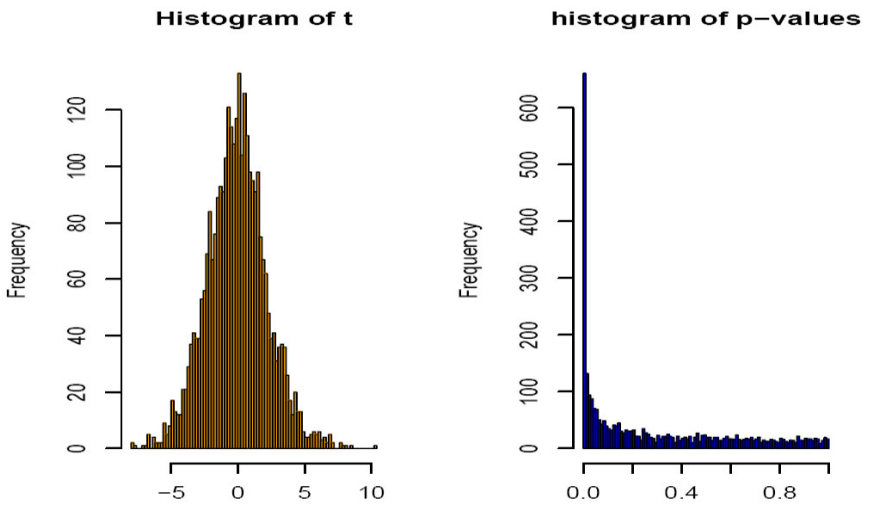


Figure 3.2: Golub data (32), 27 ALL vs. 11 AML samples, 7,074 genes. Left picture is the histogram of the calculated t-scores. Right picture is the histogram of corresponding p-values. There are 1,045 genes with p-value < 0.05.

3.1.3.1 Bonferroni correction

The Bonferroni correction is a multiple-hypotheses testing correction used when several dependent or independent statistical tests are being performed simultaneously (since while a given threshold may be appropriate for each individual testing, it is not for the set of all tests). In order to avoid a lot of false positives, the threshold needs to be lowered to account for the number of tests being performed.

The simplest and most conservative approach is the Bonferroni correction, which sets the threshold value α_{new} for the entire set of tests equal to the threshold value α of individual tests divided by the number of tests.

For the previous example when we performed the testing for 20 hypotheses,

$$\alpha_{new} = \frac{0.05}{20} = 0.0025 \tag{3.2}$$

- Pr(making a mistake) = 0.0025
- Pr(not making a mistake) = 0.9975
- Pr(not making any mistake) = 0.9975^{20} = 0.9512
- Pr(making at least one mistake) = 1 - 0.9512 = 0.0488

If we apply the proposed correction of p-values on the Golub data, we will select 98 differentially expressed genes.

When we test genes for differential expression we do two types of errors:

- False positive (Type I error): the experiment indicates that the gene has changed, but it actually has not.
- False negative (Type II error): the gene has changed, but the experiment failed to indicate the change.

Typically, researchers are more concerned about false positives because without doing many (expensive) replicates of the experiments, there will always be many false negatives.

3.1.3.2 False Discovery Rate

The false discovery rate (FDR) is the percentage of genes above a given position in the ranked list that are expected to be false positives, or percentage of selected genes that are not differentially expressed. The false positive rate (FPR) is the percentage of non-differentially expressed genes that are flagged as differentially expressed.

For the example presented in Figure 3.3, FDR and FPR have the following values:

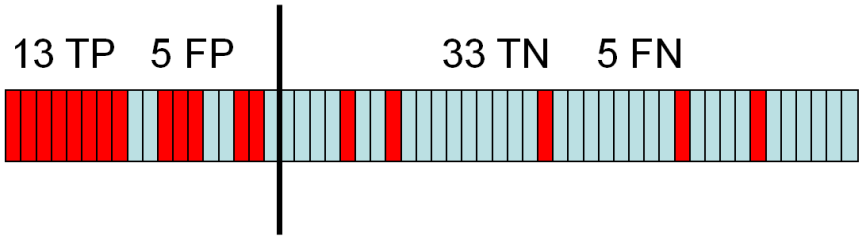


Figure 3.3: Usual scenario of Type I and Type II errors.

$$FDR = \frac{FP}{FP+TP} = \frac{5}{18} = 27.8\% \tag{3.3}$$

$$FPR = \frac{FP}{FP+TN} = \frac{5}{38} = 13.2\% \tag{3.4}$$

While estimating the FPR is harder, we can easily control the FDR by the following procedure:

- Ordered unadjusted p-values: $p_{r_1} \leq p_{r_2} \leq ... \leq p_{r_n}$, where n is the number of genes.
- To control FDR at level α, let

$$j^* = max\{j : p_{r_j} < \frac{j}{n} \cdot \alpha\} \tag{3.5}$$

- Select the genes r_j for $j = 1, ..., j^*$ as differentially expressed.

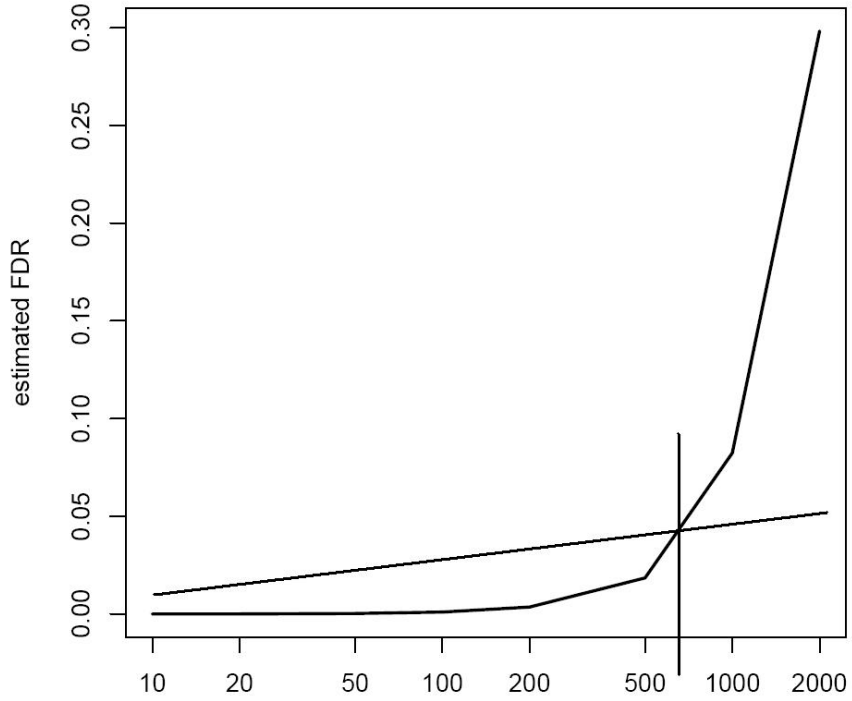


Figure 3.4: Golub data (32), 27 ALL vs. 11 AML samples, 7,074 genes. 681 genes are selected as differentially expressed.

If we apply the proposed FDR correction of p-values on the Golub data, we will select 681 differentially expressed genes. In this case, with high probability, we are sure that the number of false positives is around 35, or 5%, compared with the previous case when we did not use p-value correction, when we had around 30% false positives.

We used the procedure for finding differentially expressed genes as a test case to explain multiple testing issues. The same issues must be considered when we look for enriched gene sets. The procedure is absolutely the same, we just map the following concepts:

$$\text{gene} \equiv \text{gene set}$$
$$\text{differentially expressed gene} \equiv \text{enriched gene set}$$

Table 3.2 shows a list of most popular tools for analyzing enrichment in biologically relevant terms.

Table 3.2: Compilation of tools for functional interpretation of gene expression data. Although the most common tools have been included here, this list is not exhaustive.

Tools	Statistical Model	Correction for multiple testing	Functional labels	Site
Babelomics	Fisher's exact test, Kolmogorov-Smirnov	FDR	GO, KEGG, protein domains, location, tissues	http://www.babelomics.org
DAVID/ EASEonline	Fisher's exact test	Bonferroni	GO, pathways, diseases, protein domains	http://david.abcc.ncifcrf.gov/
FatiGO+	Fisher's exact test	FDR	GO, KEGG, protein domains	http://www.fatigo.org
GoMiner	Fisher's exact test	FDR	GO	http://discover.nci.nih.gov/gominer/
Gostat	χ^2, Fisher's exact test	FDR	GO	http://gostat.wehi.edu.au/
GO ToolBox	Hypergeometric, Fisher's exact test	Bonferroni	GO	http://gin.univ-mrs.fr/GOToolBox/
Onto-Tools	χ^2, hypergeometric, Fisher's exact test	Sidak, Bonferroni FDR	GO, KEGG	http://vortex.cs.wayne.edu/ projects.htm
FuncSpec	Hypergeometric	Bonferroni	GO, phenotypes	http://funspec.med.utoronto.ca/
GeneMerge	Hypergeometric	Bonferroni	GO, KEGG, chromosomal location	http://genemerge.bioteam.net/
GoSurfer	χ^2	Bonferroni, Sidak	GO	http://bioinformatics.bioen.uiuc.edu/ gosurfer/

In general, most of them use FDR-based multiple testing adjustments that are less conservative than Bonferroni or Sidak counterparts (14). Thus the package Babelomics (3), which includes FatiGO (1), and the Onto Tools (25; 45) would be optimal in terms of biological information content and testing strategies. DAVID/Ease (24), FunSpec (65), only for yeast, and GeneMerge (18) would be attractive from the point of view of the biological information although a bit conservative in terms of multiple testing correction. On the other hand, BayGO (82), GOMiner (90), GOstat (11), GOSurfer (91) and Ontology Traverser (88) use proper multiple-testing corrections although only provide GO terms for the annotation of the experiments. Other tools such as GO:TermFinder (16) only provide GO and are conservative in the multiple testing adjustment or even fail to provide such an adjustment.

We have shown how important are multiple testing issues in finding enriched gene sets. Any procedure that does not take this into account, as a consequence, can discover high number of spurious relationships as reliable.

3.2 Threshold-free functional interpretation

The above two-step approach is the natural choice for analyzing clusters of genes. Nevertheless, the application of a two-step strategy to the interpretation of differential gene expression in class comparison experiments causes an enormous loss of information as a large number of false negatives is accepted in order to preserve a low ratio of false positives (and the noisier the data the worse the effect). There are other limitations of the threshold-based analysis; here we list some of them:

- After correcting for multiple hypotheses testing, in selecting differentially expressed genes, a very small number of genes, or no individual gene, may meet the threshold for statistical significance, because the relevant biological differences are small relative to the noise inherent to the microarray technology.

- The opposite situation, one may be left with a long list of statistically significant genes without any common biological function, so none of the GO and KEGG terms is significantly enriched.

- Single-gene analysis may miss important effects on pathways. Biological pathways often affect sets of genes acting jointly. An increase of 20% in all genes members of a biological pathway may dramatically alter the execution of that pathway, and its impact on other processes, more than a 10-fold increase in a single gene.

- The most specific GO terms have few genes annotated so there is often not enough power to find these terms statistically significant. The more general the GO term, the more genes are annotated with it, but the less useful it is as an indication of the function of the differentially expressed genes.

To overcome these analytical challenges, recently several methods inspired from systems biology were developed. These methods focus more on collective properties of the genes than on individual gene expression values. Functionally related genes simultaneously fulfill their roles in the cell and, consequently, they are expected to display a coordinated expression. It is a long recognized fact that genes with similar overall expression often share similar functions (27; 53). This observation is consistent with the hypothesis of modularly-behaving gene programmes, where sets of genes are activated in a coordinated way to carry out functions. In this scenario, a different type of inference can be made based on testing hypotheses centered on blocks of functionally related genes, instead of testing one gene at a time.

Thus, genes can be ranked by using their differential expression values when comparing predefined classes (e.g., cases and healthy controls) by means of any appropriate statistical test (e.g., the *t*-test). The order of the genes (that cooperatively act to define pathways, functional classes) in this ranked list must be related to its participation in the distinguishing characteristic, or quality of an organism, studied in the experiment. Consequently, each functional class 'responsible' for the differences between the classes will be found in the extremes of the ranking with highest probability. Under this perspective the previous imposition of a threshold based on the rank values, which does not take into account the cooperative behavior of the genes, is thus avoided.

Figure 3.5 illustrates the threshold-free analysis strategy. Genes are arranged by differential expression between the classes N (normal) and T (test). On the right-hand side of the figure, there are labels for two different functional terms at the points in the list where genes fulfilling the corresponding roles are situated. Functional term A is completely unrelated with the experiment because different genes, belonging to this functional term, appear over-expressed in classes N and T and also in intermediate positions. Conversely, functional term B is predominantly fulfilled by genes with higher expression in class N (red values corresponding to highest expression), but scarcely appears among genes with higher expression in class T. This observation clearly points to functional term B as one of the molecular basis of the macroscopic observation made in the experiment. Instead of trying to select genes with extreme values of differential expression, systems biology-inspired methods will directly search for blocks of functionally related genes significantly cumulated in the extremes of a ranked list of genes.

A simple way of studying the asymmetrical distribution of blocks of genes across a list of ranked genes is to check if, in consecutive partitions, one of the parts is significantly enriched in any biological term with respect to their complementary part. Figure 3.5 illustrates this concept in an ordered list of genes. In this list (**C**), black circles represent genes annotated with a particular functional term and open circles represent genes with any different annotation. In the first partition, the differences (50% versus 35%), cannot be considered significant. Nevertheless, in the second partition, the differences in the

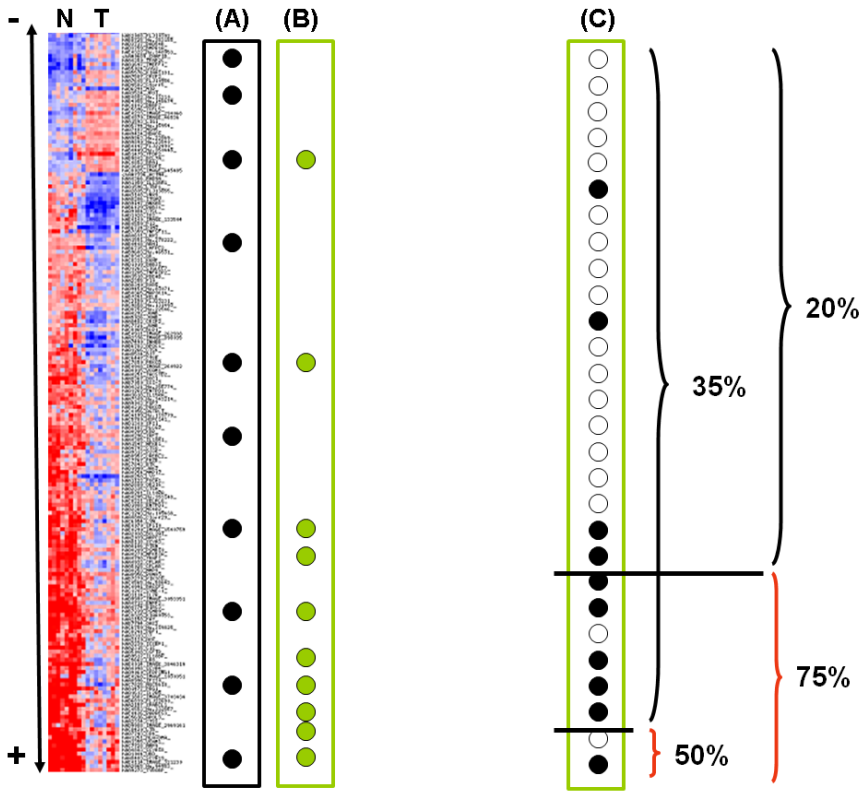


Figure 3.5: Threshold-free procedure for the functional analysis of class comparison experiments. On the left: genes ordered by differential expression between classes N (normal) and T (test). (A) Functional term unrelated to the experiment from which the rank of genes was obtained. (B) Functional term related to the experiment. (C) Schematic representation of two partitions of the segmentation test.

proportions are high enough to be declared significant (75% versus 20%): the vast majority of the genes annotated with the functional term are on the lower side of the partition.

There are different methods which have been proposed for this purpose such as the GSEA (75) or the SAFE (10) method that use a non-parametrical version of a Kolmogorov-Smirnov test. With similar accuracy, conceptually simpler and quicker methods have also been proposed such as the parametrical counterpart of the GSEA, the PAGE (47) or the segmentation test, Fatiscan (2). Here we present one representative method of both classes, GSEA and PAGE.

3.2.1 Gene Set Enrichment Analysis (GSEA)

GSEA (75) considers experiments with gene expression profiles from samples belonging to two classes. First, genes are ranked based on their t-score values. Given a predefined set of genes S (e.g., genes involved in some biological process) the goal of GSEA is to

determine whether the members of S are randomly distributed throughout ranked gene list L or primarily found at the top.

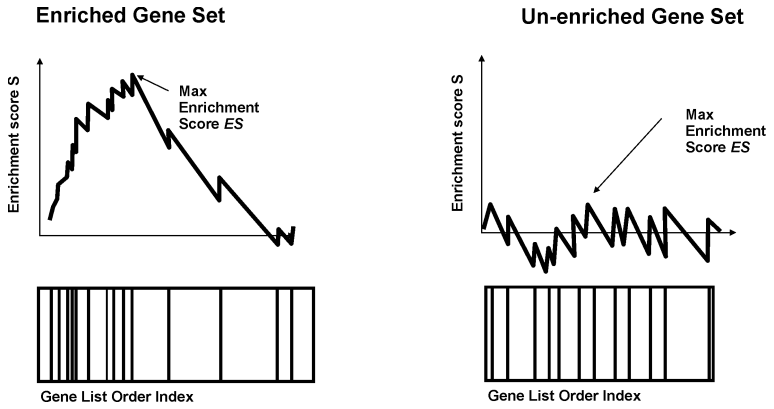


Figure 3.6: The 'spectral lines' show the positions of genes members of gene set S on the ranked gene list. This figure is borrowed from the supplementary material of (75).

There are two major steps of the GSEA method:

1. **Calculation of the enrichment score**. The enrichment score (ES) reflects the degree to which a set S is overrepresented at the top of ranked list L. The score is calculated by walking down the list L, increasing a running-sum statistic when encountering a gene in S and decreasing it when gene is not in S. The magnitude of the increment depends on the size of S, let $|S| = M$, and the total number of genes N. The enrichment score is the maximum deviation from zero encountered in the random walk (see Figure 3.6). If $L = (g_1, g_2, ..., g_N)$ is a ranked list of genes, according to their t-scores, enrichment score ES is calculated as:

$$Hit(S,i) = \sum_{\substack{g_j \in S \\ 1 \le j \le i}} \frac{1}{M} \qquad Miss(S,i) = \sum_{\substack{g_j \in S \\ 1 \le j \le i}} \frac{1}{N-M} \tag{3.6}$$

$$ES(S) = \max_{1 \le i \le N} |Hit(S,i) - Miss(S,i)| \tag{3.7}$$

2. **Estimation of the significance level of ES**. The statistical significance of the ES is computed by using an empirical phenotype-based permutation test procedure that preserves the complex correlation structure of the gene expression data. Specifically, one permutes the phenotype labels and recomputes the ES of the gene set for the permuted data, which generates a null distribution for the ES. The empirical, p-value of the observed ES is then calculated relative to this null distribution.

3.2.2 Parametric Analysis of Gene set Enrichment (PAGE)

According to the Central Limit Theorem in statistics, the distribution of the average of randomly sampled n observations tends to follow the normal distribution as the sampling size n becomes larger, even when the parent distribution from which the average is calculated is not normal. In other words, when the mean and variance of the parent distribution (whether it is normally distributed or not) are μ and σ^2, the average of n observations from the parent distribution will follow a normal distribution of mean μ and variance $\frac{\sigma^2}{n}$ when the sampling size n is large enough.

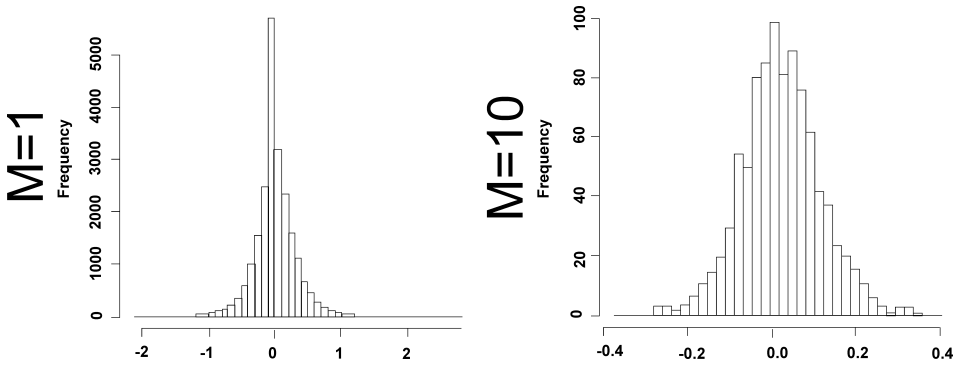


Figure 3.7: Histograms of t-score values from Golub dataset (32), when one gene is selected (left) and histogram of the average of t-scores of 10 randomly selected genes (right).

In PAGE (47), the parent distribution is a distribution of any numerical values (also termed parameters here) that describe differential expression of genes among samples in a microarray dataset. In most cases, the distribution of a parameter, i.e., the t-score values of the genes, is not normally distributed. However, as the Central Limit Theorem states, when we sample n observations from the parent distribution of a parameter, the average of the sampled observations tends to follow the normal distribution as our sampling size n becomes larger. Here, we define sampled observations as parameter values for the genes within predefined gene sets, groups of genes having similar functions, genes in the same biological pathway, and so on. If we define a gene set of sufficiently large size, e.g., 30, we can use the normal distribution to test the statistical significance of that gene set.

The following procedure is used for p value calculation of gene set S:

1. From input data containing t-score values for each gene, mean of all t-score values (μ) and standard deviation of all t-score values (σ) are calculated (this is a common step for the calculation of p-values of all genes).
2. The mean of t-scores (μ_S) of gene members of S is calculated.
3. If M is the size of S then the Z-score is calculated as
$$Z = \frac{(\mu_S - \mu) \cdot \sqrt{M}}{\sigma}$$
4. Gene set p-value is computed from the Z-score, using numerical methods (47).

3.3 Discussion

The importance of using biological information as an instrument to understand the biological roles played by genes targeted in functional genomics experiments has been highlighted in this chapter. There are situations in which the existence of noise and/or the weakness of the signal hamper the detection of over- or under-expressed genes. Improvements in methodologies of data analysis, dealing exclusively with expression values can help to some extent. Therefore, the idea of using biological knowledge as part of the analysis process is gaining popularity. In recent analysis approaches, genes are no longer the units of interest, but groups of genes with a common function. Let us consider a list of genes arranged according to their degree of differential expression between two conditions (e.g., patients versus controls). If a given biological process is accounting for the observed phenotypic differences we should then expect to find most genes involved in this process over-expressed in one of the conditions against the other. Contrarily, if the process has nothing to do with the phenotypes, the genes will be randomly distributed amongst both classes (for example if genes account for physiological functions unrelated to the disease studied, they will be active or inactive both in patients and controls). If terms were found differentially represented in the extremes of the list, one can conclude that these biological processes are significantly related to the phenotypes.

Different creative uses of information in the gene selection process as well as the availability of more detailed annotations will enhance our capability of translating experimental results into biological knowledge.

In the next chapter we present our work on integrating various sources of biological knowledge in a unified format, that was later used for the development of the new methods for functional interpretation of gene expression data.

4 Construction of an Integrated Database

Different kinds of information and data are spread over the web, hosted in a large-scale independent, heterogeneous and highly focused resources. While the time to obtain genomic data is getting shorter, the time for one to process the data and understand the biological meaning is much prolonged. Therefore, the integration of biological data and information has become an important ongoing scientific problem, as researchers still need comprehensive tools for integrative data and information processing.

There are several ongoing project that try to integrate several sources of biological knowledge and build a universal platform for the analysis of genomic data. BioWarehouse (54) is an open source toolkit for constructing bioinformatics databases using the MySQL and Oracle relational database managers. BioWarehouse integrates its component databases (ENZYME, KEGG, BioCyc, UniProt, etc.) into a common representational framework within a single database management system, thus enabling multi-database queries using the Structured Query Language (SQL). WebGestalt[1], developed at Bioinformatics Resource Center at Vanderbilt University, incorporates information from different public resources and provides an easy way for biologists to make sense out of large sets of genes. It enables biologists to manipulate integrated information and find patterns that are not detectable otherwise. WebGestalt is designed for functional genomic, proteomic and large scale genetic studies from which high-throughput data are continuously produced.

We approach this problem by dividing it in three parts: creation of a database for genomic data and information, creation of a platform for analyzing the gene expression data, and creation of a web-based tool for accessing the data and knowledge discovery. This can be done by the integration of numerous public databases (GO, KEGG Orthology, gene annotations and gene-gene interaction data) in a common, structured format, placing a broad and deep set of searchable information at the fingertips of researchers of the wider scientific community. Construction of an integrated database, described in this chapter, achieved as one of the contributions of this thesis, has been made publicly available[2].

[1] http://bioinfo.vanderbilt.edu/webgestalt
[2] http://kt.ijs.si/software/SEGS

4.1 Integration of GO and KEGG Orthology

Hierarchical relations within the set of KEGG Orthology (KO) terms suggest a model for integrating pathway data into the Gene Ontology model. Gene Ontology consists of three independent ontologies, namely Biological Process, Molecular Function and Cellular Component. We add the KEGG pathway data into the Gene Ontology data model as the fourth independent ontology of pathways (*Path*). The fourth ontology has the following GO-compliant properties.

- The set of terms in the *Path* ontology is equal to the collection of available organism-independent pathways. All terms have a unique identifier. The identifier in our model includes the prefix 'KEGG:' (KEGG:01150) to distinguish it from GO terms.
- The set of gene annotations of a pathway is the collection of genes mapped to organism-specific version of the pathway.
- Terms and annotated genes of the *Path* ontology are created independently, and hierarchically unrelated of the three remaining ontologies.
- Every parent term in the *Path* ontology implicitly includes all the annotations of its child terms.
- The top-level root term of the *Path* ontology, named 'Pathway' with the identifier KEGG:00000 holds all genes present in the KEGG pathways. This term is a placeholder, as no such general term currently exists in the KEGG database.

The proposed model makes it possible to analyze Gene Ontology terms and pathways simultaneously, and determine possible interconnections and correlation within related gene annotations. The model is not limited to the KEGG database; other pathway databases as well as different types of knowledge, such as protein-protein interactions may be integrated.

On the one hand, we recognize that viewing a pathway as an unstructured set of annotated genes greatly simplifies the picture, as we disregard internal dependencies, chemical building blocks and internal rules of behavior. On the other hand, a high-level overview of participating pathway genes with combined data of molecular functions, biological processes and cell locations may help to hypothesize about more general ideas of the biological domain.

We use a Prolog like format for representing the combined GO-KO ontology, that is also used for gene annotation database and gene-gene interaction. The general form of the representation is:

[*Object*, *List_of_properties*]

In case of the common GO-KO ontology:

- *Object* is a GO or KO identifier (GO:XXXXXXX or KEGG:XXXXX),
- *List_of_properties* is a quadruple [*ontology_id*, *term_name*, *is_a_terms*, *part_of_terms*], where *ontology_id* is one of the four identifiers: 'biological_process', 'molecular_function', 'cellular_component' or 'KEGG_pathway'. For the *Path* ontology, *part_of_terms* is always empty.

Figure 4.1 presents part of the integrated GO-KO ontology.

```
['GO:0031258', ['cellular_component', 'lamelli. membrane', ['GO:0031253', 'GO:0031256'], ['GO:0030027']]]
['GO:0000326', ['cellular_component', 'protein storage vacuole', ['GO:0000322', 'GO:0000325'], []]]
['GO:0001875', ['molecular_function', 'lipopolysac. receptor activity', ['GO:0001530', 'GO:0008329'], []]]
['GO:0003697', ['molecular_function', 'single-stranded DNA binding', ['GO:0043566'], []]]
['GO:0002262', ['biological_process', 'myeloid cell homeostasis', ['GO:0001776'], []]]
['GO:0009245', ['biological_process', 'lipid A biosynthetic process', ['GO:0046493'], ['GO:0009103']]]
['GO:0009238', ['biological_process', 'enterobactin metabolic process', ['GO:0006725', 'GO:0009237'], []]]
['KEGG:00310', ['KEGG_pathway', 'Lysine degradation', ['KEGG:01150'], []]]
['KEGG:00350', ['KEGG_pathway', 'Tyrosine metabolism', ['KEGG:01150'], []]]
```

Figure 4.1: Part of GO-KO ontology. In September 2007, the GO-KO ontology has about 22,000 terms, of which 273 are KO terms.

This database can be efficiently stored in a hash data structure supported by several programming languages (e.g., Python, Java, C++), and is also simple for parsing.

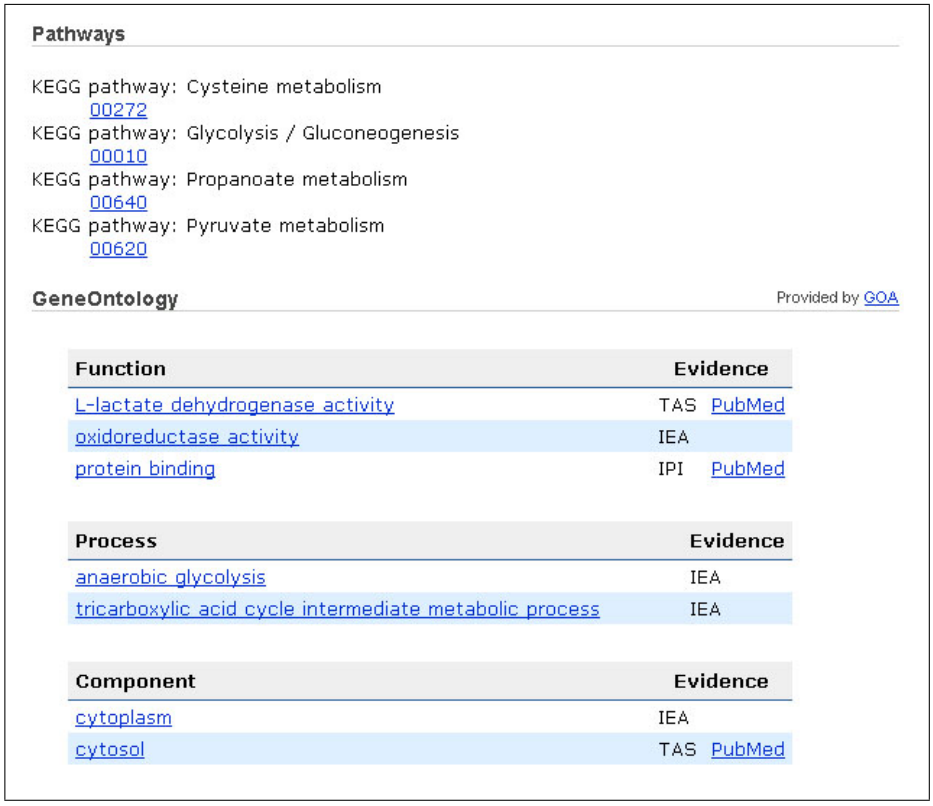
4.2 Integration of GO and KO gene annotations

The gene annotations are composed of attributes that describe the names and the structural and functional characteristics of known genes, the tissues in which the genes are expressed, the gene's protein products, the known relationship among genes, the gene's correlation with different pathologies and the biochemical pathways in which they are involved.

At present, the gene functional annotations are probably those carrying the most interesting information and their analysis could highlight new biological knowledge such as the identification of functional relationships among genes and involvement of specific genes in pathological process.

In the spirit of previous merging of GO and KO, here we also present the creation of the integrated database of gene GO-KO annotations. The original files for separate

GO and KO annotations can be found on the ENTREZ[1] and KEGG[2] site, respectively. Figure 4.2 displays part of the ENTREZ web page containing the annotations of the gene LDHA lactate dehydrogenase with KEGG and GO terms.



Pathways

KEGG pathway: Cysteine metabolism
00272
KEGG pathway: Glycolysis / Gluconeogenesis
00010
KEGG pathway: Propanoate metabolism
00640
KEGG pathway: Pyruvate metabolism
00620

GeneOntology Provided by GOA

Function	Evidence
L-lactate dehydrogenase activity	TAS PubMed
oxidoreductase activity	IEA
protein binding	IPI PubMed

Process	Evidence
anaerobic glycolysis	IEA
tricarboxylic acid cycle intermediate metabolic process	IEA

Component	Evidence
cytoplasm	IEA
cytosol	TAS PubMed

Figure 4.2: A part of data, providing annotation of the gene LDHA lactate dehydrogenase with KEGG and GO terms, contained in the ENTREZ database.

As in the previous case, the general common format of the data is:

$$[Object, List\_of\_properties]$$

where:

- *Object* is a *gene_id*, i.e., an integer, representing the gene's ENTREZ identification number,
- *List_of_properties* is a sorted list of GO and KO term identifiers that represent the annotations of gene *gene_id*.

Figure 4.3 presents part of the gene annotation database.

[1] ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2go.gz
[2] ftp://ftp.genome.jp/pub/kegg/genes/organisms/hsa/hsa_pathway.list

```
[15, ['GO:0004059', 'GO:0007623', 'GO:0008415', 'GO:0016740', 'KEGG:00380']]
[25734, ['GO:0004674', 'GO:0004713', 'GO:0005515', 'GO:0006468', 'GO:0007498']]
[72685, ['GO:0004721', 'GO:0004725', 'GO:0006457', 'GO:0006470', 'GO:0016787', 'GO:0031072']]
[814663, ['GO:0004514']]
[377841, ['GO:0016787', 'KEGG:00230', 'KEGG:00240']]
[389342, ['KEGG:03010']]
[2655449, ['GO:0004252', 'GO:0005515', 'GO:0006508']]
```

Figure 4.3: Part of gene annotation database.

4.3 Gene-gene interaction data

Protein/gene interactions assemble the molecular machines of the cell, underlie the dynamics of virtually all cellular responses and reveal functional relationships between and within regulatory modules. The sum of all such interactions defines the global regulatory network of the cell.

Microarray and proteomic platform technologies now generate large datasets of protein and genetic interactions, but these datasets vary widely in coverage, data quality, annotation and availability. The assembling and collecting gene interaction data in a consistent, well-annotated format is essential for the analysis of gene functions, investigation of system level attributes and benchmarking of high-throughput interaction studies. A number of interaction databases, including BIND[1], BioGRID[2], EcoCyc[3] and HPRD[4], provide a variety of datasets and analysis tools. ENTREZ[5] (among other functionality) is a repository for interaction datasets (BIND, BioGRID, EcoCyc and HPRD) to house and distribute comprehensive collections of physical and genetic interactions. The interaction data in the ENTREZ database is freely downloadable, interaction data is updated regularly, and downloadable files are refreshed to reflect the most recent changes.

The information about the gene interactions comes from two sources of data:

- **High-throughput experiments**. High-throughput approaches aimed at identifying novel protein and gene networks have begun to enhance hypothesis-driven biochemical and genetic approaches. These hypothesis-generating high-throughput techniques include the two-hybrid method for detecting pair-wise protein interactions (38; 80), mass spectrometric analysis of purified protein complexes (37), and the

[1]http://www.bind.ca
[2]http://www.thebiogrid.org
[3]http://www.ecocyc.org
[4]http://www.hprd.org
[5]ftp://ftp.ncbi.nlm.nih.gov/gene/GeneRIF/

synthetic genetic array and molecular barcode methods for systematic detection of synthetic lethal genetic interactions (78). The type of used method for discovering the gene-gene interaction is usually included as an *evidence code*.

- **Literature**. High-throughput datasets are filled with false positive and negative interactions. This shortfall compromises both prediction of gene/protein function and network-level analysis. The primary literature contains a vast collection of well-validated physical and genetic interactions that, while searchable through publications in PubMed, are not available in a relational database. A comprehensive set of literature-derived interactions would serve as a gold standard both for high-throughput datasets and for automated text mining approaches. Encouraged by these potential applications, significant efforts to curate interaction data from the primary literature are underway by several databases (36; 60).

In order to simplify the usage of gene-gene interaction data we created a database that unifies the data provided in the four gene-gene interaction databases (BIND, BioGRID, EcoCyc and HPRD) in a common unified format.

As in the previous cases, the unified format of the data is:

[*Object*, *List_of_properties*]

where:

- *Object* is a *gene_id*, i.e., integer, representing the gene's ENTREZ identification number,
- *List_of_properties* is a sorted list of *gene_id's*, i.e., a list of integers, representing ENTREZ gene identification numbers (*gene_id*).

Figure 4.4 presents part of the gene-gene interaction database.

```
[176 , [1404, 2192, 2199, 4060, 7130, 7143]]
[177 , [3146, 6271, 6283, 6285, 6286, 55140]]
[182 , [2353, 3725, 4242, 4301, 4851, 4853, 4854]]
[185 , [183, 409, 624, 3717, 5868, 57085, 85406]]
[8870 , [819, 4170, 5594, 5595, 7917, 8743]]
[35699 , [30977, 31469, 32268, 32504, 33882, 34665, 34685, 40585, 41397, 42928]]
[59177 , [32962, 35513, 35764, 36469, 37565, 38067, 40822, 43238]]
[855312 , [850832, 851520, 852256, 852838, 853428, 855386, 856891]]
[1157783 , [1154478]]
```

Figure 4.4: Part of gene-gene interactions database. In September 2007, the number of all gene-gene interactions was about 118,000.

4.4 Gene expression data

Recall the central dogma: DNA makes mRNA, mRNA makes protein. Genomic databases contain DNA sequences. Expression databases record measurements of mRNA levels, usually via microarrays, describing patterns of gene transcription.
Comparisons of expression patterns give clues to:

- the function and mechanism of action of gene products,
- how organisms coordinate their control over metabolic processes in different conditions - for instance yeast under aerobic or anaerobic conditions,
- the variations in mobilization of genes at different stages of the cell cycle, or of the development of an organism,
- the response to challenge by a parasite,
- the response to medications of different types and dosages, to guide effective therapy.

There exist many public microarray databases which are often accompanied with some data analysis and/or visualization tools. Here we list some of them:

- **ArrayExpress**[1] - A public repository for microarray based gene expression data maintained by the European Bioinformatics Institute.
- **Gene Expression Omnibus**[2] - A database of the US National Center for Biotechnology Information, for supporting the public use and disseminating of gene expression data.
- **Cancer Program Data Sets**[3] - provides access to datasets described in cancer program publications of the Broad Institute (created by MIT, Harvard & Whitehead Institute).
- **Standford Microarray Database**[4] - stores raw and normalized data from microarray experiments, as well as their corresponding image files.
- **Gene Expression Database (GXD)**[5] - A database of Mouse Genome Informatics.
- **Rice Expression Database**[6] - holds raw and normalized data from expression profiles obtained by the Rice Microarray Project and other research groups.

[1] http://www.ebi.ac.uk/arrayexpress/
[2] http://www.ncbi.nlm.nih.gov/geo/
[3] http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi
[4] http://genome-www.stanford.edu/microarray
[5] http://www.informatics.jax.org/mgihome/GXD/aboutGXD.shtml
[6] http://red.dna.affrc.go.jp/RED/

All three created databases (GO-KO ontology, gene annotation database and gene-gene interaction database) together with five different datasets (used in our experiments) formatted in a unified format, and provided on the thesis web site[1], are free for download and usage by wider scientific community.

From our experience, we expect that this will be of great help to biologists, medical researchers and other non-computer science trained researchers because all the data are provided in a unified format that is easy for eye reading and parsing by different script (Python, Perl, Ruby, ..., etc.) and binary/interpretable (C/C++, JAVA, Pascal, Prolog, ..., etc.) languages.

In the next chapter we present our first developed method for functional interpretation of gene expression data. It uses the methodology of Relational Subgroup Discovery (RSD) in order to find gene sets with altered expression profiles.

[1] http://kt.ijs.si/software/SEGS

5 Learning Relational Descriptions of Differentially Expressed Gene Sets

Microarrays are at the center of a revolution in biotechnology, allowing researchers to simultaneously monitor the expression of tens of thousands of genes. Independent of the platform and the analysis methods used, the result of a microarray experiment is, in most cases, a list of genes found to be differentially expressed in different types of tissues. A common challenge faced by the researchers is to translate such gene lists into a better understanding of the underlying biological phenomena.

Manual or semi-automated analysis of large-scale biological datasets typically requires biological experts with vast knowledge of many genes, to decipher the known biology accounting for genes with correlated experimental patterns. The goal is to identify the relevant 'functions', or the global cellular activities, at work in the experiment. For example, experts routinely scan gene expression clusters to see if any of the clusters are explained by a known biological function. Efficient interpretation of this data is challenging because the number and diversity of genes exceed the ability of any single researcher to track the complex relationships hidden in the datasets. However, much of the information relevant to the data is contained in the publicly available gene ontologies. Including this additional data as a knowledge source for any algorithmic strategy greatly improves the analysis.

Here we present a method to identify sets of differentially expressed genes that have functional similarity in the background knowledge formally represented with gene annotation terms from the gene ontology. The input to our algorithm is a multi-dimensional numerical dataset, representing the expression of the genes under different conditions (defining the classes of examples), GO and gene-gene interaction data. The output is a set of gene sets whose expression is significantly different for one class compared to the other classes.

The gene features extracted from public databases describe the genes in terms of their functionality and interactions with other genes. Medical experts are usually not satisfied with a separate description of every important gene, but want to know the processes that are controlled by these genes. With our algorithm we are able to find these processes by indicating the genes from the preselected list of differentially expressed genes which are included in these processes.

These goals can be achieved by using the methodology of Relational Subgroup Discovery (RSD) (83). With RSD we are able to induce sets of rules characterizing the differentially expressed genes in terms of functional knowledge extracted from the gene ontology and information about gene interactions.

5.1 Related work

While the GO based tools reviewed above enable basic analysis such as identifying a set of statistically over-represented GO terms associated with a given gene set, such analysis may be insufficient to discover frequent yet more complex ontological patterns. For example, a set of differentially expressed genes may be better characterized in terms of a logical conjunction/disjunction of GO terms presence/absence statements, rather than by simple list of frequent terms. More generally, one should also take into account the GO terms associated not only to the analyzed gene set, but also to other genes that interact with some of the analyzed genes.

The formalism of relational logic used by the RSD algorithm can capture such patterns (51; 83). Paper (8) is related to our work in that it also uses relational logic descriptions for functional discrimination of genes. A principal difference from our approach is however at least threefold. Firstly, (8) uses the inductive logic programming system Progol to search for relational ontological patterns (rules). The cover-set algorithm used by Progol is arguably inappropriate for finding a set of interesting gene subgroup descriptions as we explain later in this chapter. On the contrary, our approach is based on the weighted covering algorithm more suitable for such a task. Secondly and more importantly, the approach in (8) assumes all genes in the analyzed gene set to be of the same importance when forming the pattern descriptions. This clearly ignores the fact that certain genes are more 'interesting' than others, e.g., their expression variance across different conditions is larger. When constructing gene group descriptions, our approach deliberately devotes more attention to the 'more important' genes than to those less important. Lastly, unlike our work, (8) does not consider interactions among genes or their inclusion in gene regulatory pathways as relational properties exploitable for descriptive purposes.

Another recent paper (79) also uses relational logic for learning from genomic, proteomic and related data sources, including gene ontologies. The learning objective of (79) is however rather unrelated to ours. Whereas we attempt to compactly describe differentially expressed gene sets, (79) aims to predict protein-protein interactions.

5.2 Descriptive analysis using relational features

The fundamental idea of the proposed method is outlined in Figure 5.1. First, we construct a set of differentially expressed genes, $G_C(c)$, for every class $c \in C$. These sets can be constructed in several ways. For example: $G_C(c)$ can be the set of k ($k > 0$) most correlated genes with class c, for instance computed by Pearson's correlation. $G_C(c)$ can also be the set of best k single gene predictors, using the recall values from a microarray experiment (absent/present/marginal) as the expression value of the gene. These predictors can acquire the form such as:

$$\text{If } gene_i = present \text{ Then } class = c$$

In our experiments $G_C(c)$ was constructed using a modified version of the t-test statistics. Details about the selection mechanism used in our method are presented in Section 2.3.2.1.

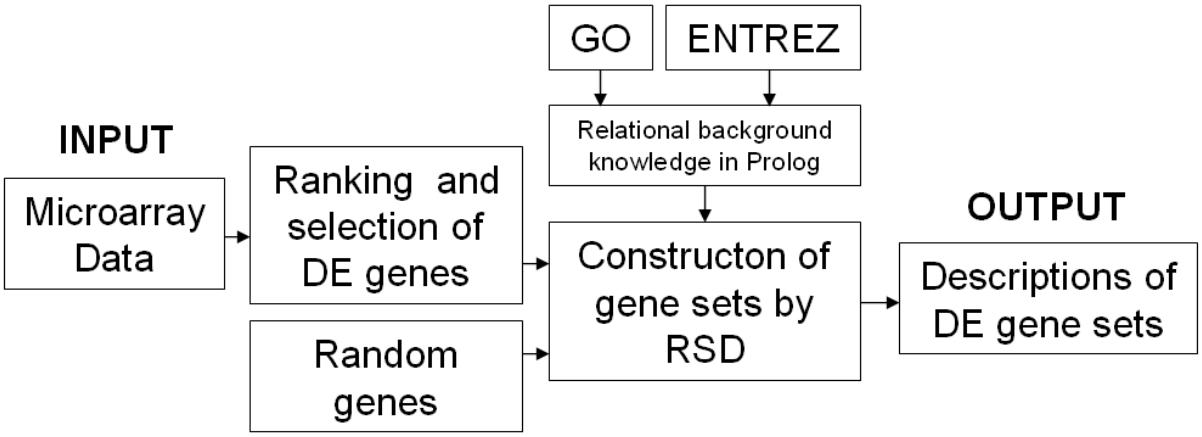


Figure 5.1: An outline of the process of microarray data analysis using RSD. First, in microarray data, we search for differentially expressed genes. Using the gene ontology information, gene annotation and gene interaction data (provided in the ENTREZ database), we produce background knowledge for differentially expressed genes on one hand, and randomly chosen genes on the other hand. The background knowledge is represented in the form of Prolog facts. Next, the RSD algorithm finds characteristic descriptions of sets of differentially expressed genes. Finally, the discovered descriptions can be straightforwardly interpreted and exploited by medical experts.

The second step aims at improving the interpretability of G_C. Informally, we do this by identifying gene sets in $G_C(c)$ (for each $c \in C$) which can be summarized in a compact way. Put differently, for each $c_i \in C$ we search for compact descriptions of gene sets with the expression strongly correlating (positively or negatively) with c_i and weakly with all $c_j \in C, j \neq i$.

Searching for these gene sets, together with their description, is defined as a separate supervised machine learning task. We refer to it as the secondary mining task, as it aims to mine from the outputs of the primary learning process in which differentially expressed genes are found. This secondary task is, in a way, orthogonal to the primary discovery process in that the original attributes (genes) now become training examples, each of which has a class label 'differentially expressed' and 'not differentially expressed'. To apply a discovery algorithm, information about relevant features of these examples is required. No such features (i.e., 'attributes' of the original attributes - genes) are usually present in the gene expression microarray datasets themselves. However, this information can be extracted from the database that we created, described in Chapter 4. For each gene we extracted its molecular functions, biological processes and cellular components where its

protein products are located, and transformed this information into the gene's *background knowledge* encoded in relational logic in the form of Prolog facts. Part of the knowledge for gene SRC, whose Entrez GeneID is 6714, is presented here:

```
function(6714,'ATP_binding').
function(6714,'receptor_activity').
process(6714,'signal_complex_formation').
process(6714,'protein_kinase_cascade').
component(6714,'integral_to_membrane').
...
```

Next, using GO, in the gene's background knowledge we also included the gene's generalized annotations. For example, if one gene is functionally annotated as: `zinc ion binding`, in the background knowledge we also included its more general functional annotations: `transition metal ion binding`, `metal ion binding`, `cation binding`, `ion binding` and `binding`. In the gene's background knowledge we also included information about the interactions of the genes, in the form of pairs of genes for which there is an evidence that they can interact:

```
interaction(6714,155).
interaction(6714,1874).
interaction(6714,8751).
interaction(6714,302).
...
```

In traditional machine learning, examples are expected to be described by a tuple of values corresponding to some predefined, fixed set of attributes. Note that a gene annotation does not straightforwardly correspond to a fixed attribute set, as it has an inherently relational character and we need to develop the relevant attributes on the basis of the pre-formed relational background knowledge. For example, a gene may be related to a variable number of cell processes, meaning it can play a role in a variable number of regulatory pathways etc. This imposes 1-to-many relations hard to elegantly capture within an attribute set of a fixed size. Furthermore, a useful piece of information about a gene g may, for instance, be expressed by the following feature involving the background knowledge of another gene:

gene g interacts with another gene whose functions include protein binding.	(5.1)

Going even further, the feature may not include only a single interaction relation but rather consider entire chains of interactions. Consequently, the task we are approaching is a case of *subgroup discovery from relational data*. For this purpose we employ the

methodology of relational subgroup discovery proposed in (51; 83) and implemented in the RSD[1] algorithm. Using RSD, we were able to discover knowledge such as:

> *Genes whose protein products are located in the nucleus, interacting with genes involved in the process of transcription regulation tend to be differentially expressed between acute myeloid leukemia and acute lymphoblastic leukemia.* (5.2)

5.2.1 The RSD algorithm

The RSD algorithm proceeds in two steps. First, it constructs a set of relational features in the form of first-order logic atom conjunctions. The entire set of features is then viewed as an attribute set, where an attribute has the value *true* for a gene (example) if the gene has the feature corresponding to the attribute. As a result, by means of relational feature construction we achieve the conversion of relational data into attribute-value descriptions.[2] In the second step, interesting gene subgroups are searched, such that each subgroup is represented as a conjunction of selected features. The subgroup discovery algorithm employed in this second step is an adaptation of the popular propositional rule learning algorithm CN2 (21).

5.2.1.1 Relational feature construction

The feature construction component of RSD aims at generating a set of relational features in the form of relational logic atom conjunctions. For example, the feature 5.1 exemplified informally in the previous section has the relational logic form:

```
interaction(A,B),function(B,'protein_binding')
```

where upper cases denote variables, and a comma between two logical literals denotes a conjunction.

The user specifies *mode declarations* which syntactically constrain the resulting set of constructed features. Each mode declaration defines a predicate that can appear in a feature, and assigns to each of its arguments a *type* and a *mode* (either input or output). Thus the following example declaration:

```
mode(3, interaction(+gene,-gene))
```

states that predicate `interaction` can appear in the feature with an input (+ sign) variable of type `gene` and an output (- sign) variable of the same type. The first declaration argument (number 3) stipulates that the predicate can appear in a single feature at most

[1] http://labe.felk.cvut.cz/~zelezny/rsd/rsd.pdf

[2] This process is known as *propositionalization* (49),(50).

3 times with the same input variable; in other words, three interactants of a single gene can be addressed in a feature.

In a feature, if two arguments have different types, they may not hold the same variable. Also, literals in a feature must be 'linked':

1. Every variable in an input argument of a literal must appear in an output argument of some preceding literal in the same feature, with the exception of the first variable in the feature (the *key* variable).

2. Inversely, every output variable of a literal must appear as an input variable of some subsequent literal.

Furthermore, the maximum length of a feature (number of contained literals) is declared, along with further optional syntactic constraints (51; 83).

Predicates with only variables in their arguments are not sufficient to capture important gene's properties. It is important that features may also contain constants (such as `'protein_binding'`). A distinguished predicate `instantiate` is used to indicate variables which will be automatically substituted by constants used in the training examples. For example, with the following declaration

```
mode(2, function(+gene,-function))
mode(1, instantiate(+function))
```

RSD first generates a constant-free feature

```
interaction(A,B), function(B,C), instantiate(C)
```

and then replaces it with a *set* of features, in each of which variable `C` is replaced by a constant and the `instantiate` predicate is removed. An example feature set consists of the following two features:

```
interaction(A,B), function(B,'protein binding')
```

and

```
interaction(A,B), function(B,'binding')
```

However, only such replacements for `C` are considered that make the resulting feature hold true for at least a pre-specified number of genes, according to a pre-specified minimal support threshold of RSD.

Given a set of declarations, RSD proceeds in the manner described above to produce an exhaustive set of features satisfying the declarations. Technically, this is implemented as an exhaustive depth-first backtrack search in the space of all feature descriptions, equipped with certain pruning mechanisms. Besides the language declarations, each feature must also comply to the *connectivity* requirement, according to which no feature may be decomposable into a conjunction of two or more features. For example, the following expression does not form an admissible feature:

```
interaction(A,B),function(B,'protein_binding'),
interaction(A,C),component(C,'membrane')
```

The reason is that it can be decomposed into two separate features, consisting of the first two (last two, respectively) literals. We do not construct such decomposable expressions, as these are clearly redundant for the purpose of subsequent search for rules with conjunctive antecedents. Note that decomposable features may in general be made undecomposable by adding a literal, such as by adding `interaction(B,C)` to the expression exemplified above. It is primarily the concept of undecomposability that allows for extensive search space pruning (51; 83) in the feature construction process.

Some examples of features constructed by RSD are listed below:

```
f(7,A):-function(A,'kisspeptin_receptor_binding').
f(8,A):-function(A,'phosphopant_binding').
f(11,A):-process(A,'intestinal_lipid_catabolism').
f(14,A):-process(A,'neurite morphogenesis').
f(19,A):-component(A,'nucleus').
f(22,A):-interaction(A,B),function(B,'mannokinase_activity').
f(24,A):-interaction(A,B),function(B,'enzyme_regulator_activity'),
                          component(B,'membrane').
f(84,A):-interaction(A,B),process(A,'glycolate_catabolism'),
                          component(B,'intrinsic_to_membrane').
```

where the 'head' of the feature definition formally indicates the feature number and the key variable.

Finally, to evaluate the truth value of each feature for each example for generating the attribute-value representation of the relational data, the first-order logic resolution procedure is used, provided by a standard Prolog language interpreter.

5.2.1.2 Subgroup Discovery

Subgroup discovery aims at finding population subgroups that are statistically 'most interesting', e.g., are as large as possible and have the most unusual statistical characteristics with respect to the property of interest (87) (see Figure 5.2).

Notice an important aspect of the above definition: there is a predefined property of interest, meaning that a subgroup discovery task aims at characterizing population subgroups of a given *target* class. This property indicates that standard classification rule learning algorithms could be used for solving the task. However, while the goal of classification rule learning is to generate predictive models in the form of rule sets that discriminate between the target class and non-target classes, subgroup discovery aims at discovering a set of individual patterns (rules) characterizing the target class.

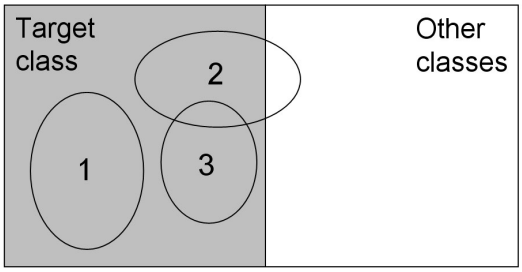


Figure 5.2: Descriptions of discovered subgroups ideally cover just individuals of the target class (subgroups 1 and 3), however they may cover also a few individuals of other classes (subgroup 2).

Rule learning typically involves two main procedures: the search procedure used to induce a single rule (see Section 5.2.1.3 below) and the control procedure (the covering algorithm) that repeatedly executes the search to induce a set of rules (see Section 5.2.1.4).

5.2.1.3 Inducing a single subgroup describing rule

The RSD algorithm (51; 83) is based on an adaptation of the standard propositional rule learner CN2 (21). Its search procedure used in learning a single rule performs beam search, starting from the empty conjunct, successively adding conditions (relational features). In CN2, classification accuracy of a rule is used as a heuristic function in the beam search. The accuracy[1] of an induced rule of the form $H \leftarrow B$ (where H in the rule head is the target class, and B is the rule body formed by a conjunction of relational features) is equal to the conditional probability of head H, given that body B is satisfied: $p(H|B)$.

In RSD, the accuracy heuristic $Acc(H \leftarrow B) = p(H|B)$ is replaced by the *weighted relative accuracy* heuristic. Weighted relative accuracy is a reformulation of the Piatetsky-Shapiro heuristics used in MIDOS (87), aimed at balancing the size of a group with its distributional unusualness (48). It is defined as follows:

$$WRAcc(H \leftarrow B) = p(B) \cdot (p(H|B) - p(H)). \tag{5.3}$$

Weighted relative accuracy consists of two components: generality $p(B)$, and relative accuracy $p(H|B) - p(H)$. The second term, relative accuracy, is the accuracy gain relative to the rule $H \leftarrow true$, which predicts all instances to satisfy H. Hence, rule $H \leftarrow B$ is only interesting if it improves upon this 'default' accuracy. Another way of viewing relative accuracy is that it measures the utility of connecting rule body B with rule head H. Note that it is easy to obtain high relative accuracy with very specific rules, i.e., rules with low generality $p(B)$. To this end, generality is used as a 'weight' which trades off generality of the rule (rule coverage $p(B)$) and relative accuracy $(p(H|B) - p(H))$.

[1] In some contexts, this quantity is called *precision*.

In the computation of *Acc* and *WRAcc* all probabilities are estimated by relative frequencies[1] as follows:

$$Acc(H \leftarrow B) = p(H|B) = \frac{p(HB)}{p(B)} = \frac{n(HB)}{n(B)} \tag{5.4}$$

$$WRAcc(H \leftarrow B) = \frac{n(B)}{N} \cdot \left(\frac{n(HB)}{n(B)} - \frac{n(H)}{N} \right) \tag{5.5}$$

where N is the number of all the examples, $n(B)$ the number of examples covered by rule $H \leftarrow B$, $n(H)$ the number of examples of class H, and $n(HB)$ the number of examples of class H correctly classified by the rule (true positives).

An example subgroup describing rule, constructed as conjunction of two features (numbered 81 and 254), is given below:

```
subgroup(A) = f(81, A), f(254, A),
```

where

```
f(81, A) = interaction(A,B), process(B,'phosphorylation')
```

and

```
f(254, A) = interaction(A,B), process(B,'negative regulation of apoptosis'),
component(B,'intracellular membrane-bound organelle')
```

5.2.1.4 Inducing a set of subgroup describing rules

In CN2, for a given class in the rule head, the rule with the best value of the heuristic function found in the beam search is kept. The algorithm then removes all examples of the target class satisfying the rule's conditions (i.e., positive examples *covered* by the rule) and invokes a new rule learning iteration on the remaining training set. All negative examples (i.e., examples that belong to other classes) remain in the training set.

In this classical covering algorithm, only the first few induced rules may be of interest as subgroup descriptors with sufficient coverage, since subsequently induced rules are induced from biased example subsets, i.e., subsets including only positive examples not covered by previously induced rules. This bias constrains the population of individuals in a way that is unnatural for the subgroup discovery process, which is aimed at discovering characteristic properties of subgroups of the target population.

[1] Alternatively, the Laplace (20) and the m-estimate (19) could also be used.

In contrast, RSD uses the *weighted covering algorithm*, which allows for discovering interesting subgroup properties in the entire target population. The weighted covering algorithm modifies the classical covering algorithm in such a way that covered positive examples are not deleted from the set of examples to be used to construct the next rule. Instead, in each run of the covering loop, the algorithm stores with each example a count that indicates how many times (with how many induced rules) the example has been covered so far.

By default, initial weights of all examples e_j are set to 1 (alternatively, as was the case in our experiments, the initial weights of the examples may encode the apriori importance of a given example). In subsequent iterations of the weighted covering algorithm all target class examples weights decrease according to the formula $\frac{1}{i+1}$, where i is the number of constructed rules that cover example e_j. In this way the target class examples whose weights have not been decreased will have a greater chance to be covered in the following iterations of the weighted covering algorithm.

The combination of the weighted covering algorithm with the weighted relative accuracy thus implies the use of the following *modified WRAcc* heuristic:

$$WRAcc(H \leftarrow B) = \frac{n'(B)}{N'} \cdot \left(\frac{n'(HB)}{n'(B)} - \frac{n(H)}{N} \right) \tag{5.6}$$

where N is the number of examples, N' the sum of the weights of all examples, $n(H)$ the number of examples of class H, $n'(B)$ the sum of the weights of all covered examples, and $n'(HB)$ the sum of the weights of all correctly covered examples.

An example set of rules is given below:

```
subgroup1(A) = f(81, A), f(254, A),
subgroup2(A) = f(34, A), f(103, A),
subgroup3(A) = f(54, A), f(180, A)
```

where

```
f(81, A) = interaction(A,B), process(B,'phosphorylation')
f(254, A) = interaction(A,B), process(B,'negative regulation of apoptosis'),
component(B,'intracellular membrane-bound organelle')
f(34, A) = interaction(A,B),function(B,'metal ion binding'),
component(B,'membrane')
f(103, A) = interaction(A,B),function(B,'struct.  constit.  of cytoskeleton')
f(54, A) = interaction(A,B),function(B,'metal ion binding'),
process(B,'transcription, DNA-dependent')
f(180, A) = interaction(A,B),process(B,'reg.  of transcript., DNA-dependent').
```

5.3 Experiments

In this section we present the experiments and analysis of the results used for demonstrating the applicability of the new developed methodology.

5.3.1 Materials and methods

We apply the proposed methodology on three classification problems from gene expression data, with the aim to describe the genes that are usually used by the classifiers, i.e, the differentially expressed genes.

The first problem was introduced in (32) and aims at distinguishing between samples of ALL and AML from gene expression profiles obtained by the Affymetrix HU6800 microarray chip, containing probes for 6817 genes. The data contains 73 class-labeled samples of expression vectors. The second problem was described in (66) and aims at distinguishing different subtypes of ALL (6 recognized subtypes plus a separate class 'other' containing the remaining samples). The data contains 132 class-labeled samples obtained by Affymetrix HG-U133 set of microarrays, containing 22,283 probes. The third problem was defined in (64). Here one tries to distinguish among 14 classes of cancers from gene expression profiles obtained by the Affymetrix Hu6800 and Hu35KsubA microarray chip, containing probes for 16,063 genes. The dataset contains 198 class-labeled samples. Note that out method does not address the learning task of discriminating between the classes. Instead, for the given target class we aim at finding the most characteristic description of its differentially expressed genes.

To access the annotation data for every gene considered, it was necessary to obtain unique gene identifiers from the microarray probe identifiers available in the original data. We achieved this by script-based querying of the Affymetrix site[1] for translating probe ID's into unique gene ID's. Knowing the gene identifiers, information about gene annotations and gene interactions can be extracted from the ENTREZ, that is included in our database. We developed a program script in the Python language, which extracts gene annotations and gene interactions from our database, and produces their structured, relational logic representations which can be used as input to RSD.

For all three datasets, and for each class c we first extracted a set of differentially expressed genes $G_C(c)$. In our experiments we used t-test score $T(g, c)$ for selecting differentially expressed genes. t-test is a test of the null hypothesis that the means of two normally distributed populations are equal. Higher $|T(g, c)|$ means lower probability which in turn means that mean gene expression is different between different classes.

[1] www.affymetrix.com/analysis/netaffx/

$T(g, c)$ is computed by the following formula:

$$T(g, c) = \frac{\mu_1(g) - \mu_2(g)}{\sqrt{\frac{\sigma_1(g)}{N_1} + \frac{\sigma_2(g)}{N_2}}} \tag{5.7}$$

where $N_1 = |c|$, $N_2 = |C \setminus c|$, $[\mu_1(g), \sigma_1(g)]$ and $[\mu_2(g), \sigma_2(g)]$ denote the means and standard deviations of the logarithm of the expression levels of gene g for the samples in class c and samples in $C \setminus c$, respectively.

$T(g, c)$ reflects the difference between the classes relative to the standard deviation within the classes. Large values of $|T(g, c)|$ indicate a strong correlation between the expression of gene g and class c, while the sign of $T(g, c)$ being positive (negative) corresponds to g being highly (less) expressed in class c than in the other classes. Unlike a standard Pearson's correlation coefficient, $T(g, c)$ is not confined to the range $[-1, +1]$. In order to avoid situations illustrated in Figure 5.3, where genes B and C would have similar values of $|T(g, c)|$ but where C is not significantly differentially expressed, we dictate one more condition for a gene to be selected: $|\mu_1(g) - \mu_2(g))| > 1$. Thereby we ensure that selected genes have at least twofold difference in their average expression for the given class.

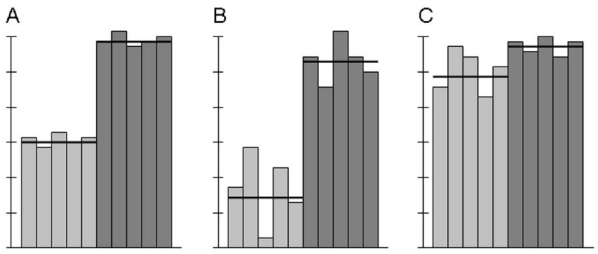


Figure 5.3: Expression of three genes (A, B and C) for five patients of class 1 and five patients of class 2. Perfect class distinction can be achieved by idealized gene A, in which the expression level is uniformly low in class 1 and uniformly high in class 2. A more realistic case is gene B which is also useful for class distinction. We do not use gene C for class distinction as we are interested in genes that have significant difference in their mean expression between the classes.

For all three problems and all classes we selected the 50 most differentially expressed (highest t-score ranking) genes and the same number of randomly chosen non-differentially expressed genes. The specific number of selected genes is a matter of trade-off. Including a high number of examples in the training set is in general preferable for learning. However, extending the training set to relatively low-scoring genes decreases the overall quality of the training set. A full quantification of this trade-off is out of the scope of this study, where we adhere to 50 examples of each class. This is a usual number of selected genes in the context of microarray data classification with support vector machines or voting algorithms (32).

The average, maximal and minimal values of $|T(g, c)|$ for the selected differentially expressed genes for each problem/class are listed in Table 5.1. In general, higher numbers mean that the class is easier to distinguish from the other classes on the basis of the expression of the most differentially expressed genes.

Table 5.1: Average, maximal and minimal value of $|T(g, c)|$, for $g \in G_C(c)$, for each problem and class c.

Task	Class	Avg	Max	Min
ALL-AML	ALL	6.74	11.09	5.31
	AML	6.74	11.09	5.31
Subtypes	BCR	5.95	10.30	4.65
of ALL	E2A	11.68	38.80	8.46
	HD50	6.09	8.56	5.21
	MLL	8.71	13.15	6.85
	T_ALL	16.70	27.12	12.66
	TEL	9.69	17.59	7.34
Multy	BREAST	6.53	8.42	5.86
class	PROSTATE	6.05	11.90	4.84
	LUNG	5.04	8.56	4.25
	COLORECTAL	5.71	14.83	4.42
	LYMPHOMA	8.73	14.69	7.32
	BLADDER	5.91	10.27	5.07
	MELANOMA	6.53	11.28	5.71
	UTERUS	5.07	7.49	4.46
	LEUKEMIA	11.55	17.02	9.78
	RENAL	4.65	6.62	4.06
	PANCREAS	5.22	7.92	4.32
	OVARY	4.06	6.33	3.59
	MESOTHELIOMA	4.81	9.51	4.61
	CNS	11.99	23.06	9.47

The usage of the gene t-test score $T(g, c)$ is twofold. In the first part of the analysis it is used for the selection of differentially expressed genes as described above. Secondly, it acts as the initial weight for each example gene in the subgroup discovery procedure where we try to characterize these differentially expressed genes. In this secondary mining task, RSD will thus prefer to group genes with large weights. As a consequence, such important genes are typically covered by more than one reported subgroup description, each time with an alternative description.

5.3.2 Experimental results

To illustrate the straightforward interpretability of the induced gene set descriptions, we use as an example the best-scoring gene subgroups discovered by RSD for the CNS (central nervous system) and breast cancer class from the 14-class cancer problem.

A group of genes, `CSN-geneSet(A)`, differentially expressed between CNS on one hand and the other classes, was defined by RSD through the conjunction of two relational logic features:

```
CNS-geneSet(A) = f(81, A), f(254, A),
```

where

```
f(81, A) = interaction(A,B), process(B,'phosphorylation')
```

and

```
f(254, A) = interaction(A,B), process(B,'negative regulation of apoptosis'),
component(B,'intracellular membrane-bound organelle')
```

This gene group, defined by the interaction with genes involved in phosphorylation, negative regulation of apoptosis and intracellular localization, contains 7 differentially expressed genes and none of the non-differentially expressed genes used as the negative examples by the algorithm. The gene group members are brain specific genes and genes active in cellular survival. The former includes glial fibrillar astrocytic protein [GFAP, 2670] and reticulon 4 [neurite growth factor, 57142] exhibited positive expression scores as would be predicted in brain derived cancers. The latter, cell death genes caspase 4 [837] and tumor necrosis factor receptor type I associated death domain protein [TRADD, 8717] are both associated with decreased expression, also an expected finding, as lower levels of these cell death/pro-apoptotic genes are associated with uncontrolled cellular growth in malignancy and are one of the most prominent features of cancers.

These observations support the validity of our method (as they fit biological expectations based on scientific and clinical investigations unrelated to ours) and thus give credibility to findings related to the remaining genes in the subgroup, of which little is known in brain cancers. These include glycogen synthase kinase 3 beta and nuclear receptor corepressor 2. Glycogen synthase kinase 3 beta is a master switch of multiple processes involved in cellular biology by definition exercising its regulatory effects by phosphorylation. Specifically it is critical for cell migration, proliferation (including pathological cellular proliferation in multiple human cancers) and it has been previously reported to be functionally connected to brain protein tau (89). To our knowledge, however, nothing is known of its

role on brain tumors. The role of the nuclear receptor corepressor 2 has been described for breast and prostate cancer. As their role in brain cancer is not known and based on our data their expression is indeed significantly increased in brain tumors (when compared to other malignancies) the nuclear receptor corepressor 2 and glycogen synthase kinase 3 beta represent good candidate genes for further investigations in etiology of brain cancer.

A group of genes, `breast-geneSet(A)`, differentially expressed (in this case it was under-expressed) between *breast* on one hand and the other classes, was defined by RSD through the conjunction of two relational logic features:

```
breast-geneSet(A) = f(14, A), f(38, A),
```

where

```
f(14, A) = process(A,'regulation of transcription')
```

and

```
f(38, A) = function(A,'zinc ion binding')
```

This gene group is made of genes involved in regulation of transcription and in zinc ion binding. Zinc is a cofactor in protein-DNA binding, via a 'zinc finger' domain. This property is shared by many transcription factors, which are major regulators of normal and abnormal (e.g., malignant) cell proliferation, therefore '`regulation of transcription`' was not found interesting. Second, zinc is an essential growth factor. Less than optimal expression of the factors involved in zinc metabolism can therefore represent either a cause or a biomarker of dysregulated cellular proliferation. By combining the both features, '`regulation of transcription`' and '`zinc ion binding`' RSD was able to construct gene set that was composed of mostly differentially expressed genes.

5.3.3 Statistical validation

Here we present a statistical validation of the proposed methodology for discovering descriptions of differentially expressed gene sets. Specifically we wish to determine if the high descriptive capacity pertaining to the incorporation of the expressive relational logic language incurs a risk of *descriptive overfitting*, i.e., a risk of discovering subgroups whose bias toward differential expression is only due to chance. We thus aim at measuring the discrepancy of the quality of discovered subgroups on the training data on one hand and independent test sets on the other hand, as performed by 5-fold stratified cross-validation[1].

[1]Same as cross-validation, except that the folds are stratified so that they contain approximately the same proportions of labels as the original dataset.

The specific qualities measured for each set of subgroups produced for a given class are average *precision* (PRE), *recall* (REC) and *area under ROC* (AUC)[1] values among all subgroups in the subgroup set. Table 5.2 shows the PRE and REC values results for the three respective problem domains[2].

Table 5.2: Precision, recall and AUC figures of found subgroups, for the set of ALL/AML, Subtypes of ALL and Multi-Class-Cancer differentially expressed genes, obtained through 5-fold cross-validation.

Task	Data	PRE	REC	AUC
ALL-AML	Train	100(±0)%	16%	65%
	Test	85(±6)%	13%	60%
Subtypes	Train	95(±4)%	17%	63%
of ALL	Test	78(±10)%	12%	61%
Multy	Train	94(±6)%	14%	59%
class	Test	75(±12)%	12%	57%

Overall, the results demonstrate an acceptable decay from the training to the testing set in terms of both PRE and REC, suggesting that the discovered subgroup descriptions indeed capture the relevant gene properties. In terms of *total* coverage, in average, RSD covered more then $\frac{2}{3}$ of the preselected differentially expressed genes, while $\frac{1}{3}$ of the preselected genes were not included in any group. A possible interpretation is that they are not functionally connected with the other genes and their initial selection through the t-test was due to chance. This information can evidently be back-translated into the gene selection procedure and used as a gene selection heuristic. This approach is out of the scope of the thesis but represents a direction for future work.

The risk of descriptive overfitting suggested by the results of Table 5.2 is due to two reasons: first, the imperfections in the data and second, the high expressiveness of the relational logic language.

Concerning the first reason, the existing gene annotation databases are currently rather coarse-grained in that high-confidence classification of genes into low-level (i.e., specific) ontological classes is rarely available. A second source of input imperfectness is the fact that functions, locations and involved processes are known for only a subset of genes. Furthermore, most annotation databases are built by curators who manually review the existing literature. It is thus possible that certain known facts get temporarily overlooked.

[1]Definitions of PRE, REC and AUC can be found in (83).

[2]For the first problem we had one set of differentially expressed genes, where for the second (third) problem we had 6 (14) sets of differentially expressed genes and equal number of learning tasks, one for each class, where results of each subtask were averaged.

For instance, (45) found references in the literature published in the early 90s, for 65 functional annotations that are not yet included in the current functional annotation databases.

Secondly, the language expressivity allows for forming rather complex rules, involving both gene-ontological terms and gene-interaction relations. As such they are possibly prone to capturing noise in data rather than genuine biological principles.

Despite the two described factors, the overfitting effect manifests itself to an acceptable extent and the rule quality measured on independent testing sets is still relatively high. Moreover, some of the actual discovered patterns also lead to biologically plausible interpretations as demonstrated in Section 5.3.2.

5.3.4 Analyzing individual components of the methodology

We further experimented with different settings of our algorithm in order to investigate the influence of different ingredients of the approach on the precision of the found descriptions. In addition to the original setting (ORIG), we performed experiments with three alternative settings: without gene-interaction information (-INTERACTION), without GO term generalization (-GO), and without incorporating gene t-test scores as the initial weights in the RSD's weighted covering algorithm for subgroup discovery, thus initializing all weights to 1 (-WEIGHT). In Table 5.3 we present the test-set results averaged in 5-fold cross-validation.

Table 5.3: Precision of discovered differentially expressed gene group descriptions, for three scenarios where part of the background knowledge or gene-weight information was removed.

TASK	ORIG	-INTERACTION	-GO	-WEIGHTS
ALL-AML	85(±6)%	44(±12)%	72(±13)%	75(±8)%
Subtypes of ALL	78(±10)%	52(±13)%	74(±16)%	71(±12)%
Multy class	75(±12)%	45(±16)%	56(±14)%	73(±14)%

Table 5.3 shows that all the three ingredients exhibit a strong positive influence on the results, with interaction data being the strongest factor.

5.4 Discussion

In this chapter we presented a method that uses gene ontologies, together with the paradigm of relational subgroup discovery, to help find patterns of expression for genes with a common biological function that correlate with the underlying biology responsible for class differentiation. Our methodology proposes to first select a set of important differentially expressed genes for all classes and then find compact, relational descriptions of subgroups among these genes.

It is noteworthy that the latter descriptive 'post-processing' step is a machine learning task, in which the curse of dimensionality usually ascribed to microarray data classification, actually turns into an advantage. This is because, in traditional microarray data mining configurations, the high number of genes results in a high number of attributes usually confronted with a relatively small number of expression samples, thus forming grounds for overfitting. In our approach, on the contrary, genes correspond to examples and thus their abundance is beneficial. Furthermore, the dimensionality of the secondary attributes (relational features of genes extracted from gene annotations) can be conveniently controlled via suitable constraints of the language grammar used for the automatic construction of the gene features.

A further remark concerns the fact that genes are frequently associated to multiple functions, i.e., they may under some conditions exhibit a behavior of genes with one function while in other conditions a different aspect of their function may be important. Here the subgroup discovery methodology is effective at selecting a specific function important for the classification. Indeed, one given gene can be included in multiple subgroup descriptions (this was e.g., the case of genes with id's 51592 and 115426 in the breast cancer class), each emphasizing the different biological process critical to the explanation of the underlying biology responsible for the observed experimental results.

Yet another aspect of the proposed method is of interest, following from the illustrative example of a discovery result provided in Section 5.3.2. Here the discovered subgroup contains four genes whose differential expression (for the CNS cancer class) is well in accordance with the biological state of the art. The group is described using the *features shared by the genes*, rather than through plain gene list as in traditional approaches. As a consequence, the group also includes further genes sharing the features, whose connection to brain cancer has not yet been described, yet closer analysis reveals evidence that such association is indeed plausible. We believe that this 'generalization' aspect of the proposed methodology may contribute to discovering new marker genes by proposing candidate genes for further experimental evaluation.

We have assessed the quality of the induced descriptions by evaluating them on independent test sets using 5-fold cross-validation. The results show a clear advantage of using all the complementary sources of background knowledge in the description generation procedure (GO ontology, gene interactions as well as degree of differential expression of genes represented by gene weights), as shown in Table 5.3.

We believe that the presented approach can significantly contribute to the application of relational machine learning to gene expression analysis. Despite the demonstrated benefits of the methodology, the precision and recall evaluation of descriptors in Table 5.2 suggests that there is still room for improvement. This is to be achieved through the expected increase in both the quality and quantity of gene/protein annotations in the near future.

In the next chapter we present our second developed method for functional interpretation of gene expression data. The main component of the developed method is an efficient algorithm for the construction of new biologically interesting gene sets. After the construction, the gene sets are tested for enrichment by the standard methods for enrichment analysis.

6 SEGS: Search for Enriched Gene Sets

Gene Ontology (GO) terms are often used to interpret the results of microarray experiments. The most common approach is to perform Fisher's exact test (57) to find gene sets annotated by GO terms which are over-represented among the genes declared to be differentially expressed in the analysis of microarray data. Another way is to apply Gene Set Enrichment Analysis (GSEA) (75) that uses predefined gene sets and ranks of genes to identify significant biological changes in microarray datasets. However, after correcting for multiple hypotheses testing, few (or no) GO terms may meet the threshold for statistical significance, because the relevant biological differences are small relative to the noise inherent to the microarray technology.

In addition to the individual GO terms, we propose testing of gene sets constructed as intersections of GO terms, Kyoto Encyclopedia of Genes and Genomes Orthology (KO) terms, and gene sets constructed by using gene-gene interaction data obtained from the ENTREZ database. Our method finds gene sets that are significantly over-represented among differentially expressed genes which can not be found by the standard enrichment testing methods applied on individual GO and KO terms, thus improving the enrichment analysis of microarray data.

6.1 Related work

Tests for gene set enrichment compare lists of differentially expressed (DE) genes and non-DE genes to find which gene sets annotated by GO and KO terms are over- or under-represented amongst the DE genes. Several research groups have developed software to carry out Fisher's exact tests to find which gene sets are over-represented among the genes found to be differentially expressed, e.g., (1; 11) and other works cited in (45). The Fisher's test for term T essentially compares the proportion of DE genes annotated by term T with the proportion of non-DE genes annotated by term T. Since there is a test for each of several thousands of GO nodes, and hundreds of KO nodes, multiple hypothesis testing must be taken into account. This is usually done by the Bonferroni correction or a more sophisticated correction controlling the False Discovery Rate (FDR). Benjamini and Hochberg's method (13) gives valid control of the FDR even when the different tests are dependent.

Approaches based on Fisher's exact testing have some major limitations:

- After correcting for multiple hypothesis testing, in selecting DE genes, no individual gene may meet the threshold for statistical significance, because the relevant biological differences are small relative to the inherent microarray technology noise.

- The opposite situation, one may be left with a long list of statistically significant genes without any common biological function, so none of the gene sets annotated by GO and KO terms is significantly enriched.

- Single gene analysis may miss important effects on pathways. Biological pathways often affect sets of genes acting jointly. An increase of 20% in the expression of all gene members of a biological pathway can alter the execution of that pathway, and its impact on other processes, significantly more than a 10-fold increase in a single gene (63).

- It is not rare that different research groups studying the same biological system report lists of DE genes they found to be statistically significant which have just a small overlap (28).

- Since all genes annotated by a given GO term are also annotated by all of its parents, closely related nodes may be found separately significant (4).

- Specific GO terms have few genes annotated, so there is often not enough statistical evidence to find these terms as statistically significant. The more general the GO term, the more genes are annotated by it, but the less useful the term is as an indication of the function of the differentially expressed genes (55).

Several methods have been developed recently to overcome the presented analytical challenges. For improving the sensitivity of enrichment detection, Gene Set Enrichment Analysis (GSEA) (75) and Parametric Analysis of Gene Set Enrichment (PAGE) (47) were developed. GSEA calculates an enrichment score (ES) for a given gene set using ranks of genes and infers the statistical significance of ES against the ES-background distribution calculated by permuting the labels of the original dataset. In the new version of GSEA, GSEA-P (76), there is an option for importing gene sets from MSigDB (Molecular Signatures Database) and testing them for enrichment, thus increasing the probability for finding enriched gene sets. In contrast, PAGE calculates a Z-score for a given gene set from a parameter such as t-score value calculated on the basis of two experimental groups and infers statistical significance of the Z-score against the standard normal distribution. These two methods are capable to find enriched gene sets, not detectable by the standard Fisher's exact test.

(34) take into account the hierarchical structure of the GO by measuring the over-representation of each term relative to its parent terms. (4) downweight the contribution of genes to the calculation of over-representation of a term if the children of that term have already been found significantly enriched. These two methods do not improve the statistical power, as the number of genes in each hypothesis test will be smaller than in the usual term-by-term tests, as double counting is penalized. However, they do help to improve the interpretation, since they produce just one (or at least not too many) significant p-values for each significant region of the graph. (55) use grouping of similar GO terms (which are close in the GO graph) in order to increase the statistical power. The reason is that the lower terms in the GO have few genes annotated by it, and can not be found statistically significantly enriched. Therefore, (55) group several terms to increase the size of the gene sets tested for enrichment. This approach is useful and can find enriched gene sets not detectable by standard screening of GO terms, but it is different form ours: we construct new gene sets as intersections of gene sets defined by Molecular Function, Biological Processes and Cellular Component terms of GO and KO terms, whereas (55) create new gene sets by making union of similar terms in GO. Concerning the usage of KO term in enrichment analysis, the work of (58) uses KO terms for automated annotation of large sets of genes, including whole genomes, and automated identification of pathways. This is done by identifying both the most frequent and the statistically significantly enriched pathways.

6.2 The proposed SEGS approach

In this thesis we propose a novel approach for searching of enriched gene sets (SEGS) which proves to further improve the gene set enrichment results and by that the interpretation of gene expression data. Our approach is based on the efficient generation of new

biologically relevant gene sets, that are tested for possible enrichment. The new gene sets are generated as intersections of GO and KO terms and gene sets defined with the help of gene-gene interaction data. Testing the enrichment of these gene sets with the standard methods (Fisher's exact test, GSEA and PAGE) shows that our method finds gene sets constructed from GO and KO terms significantly over-represented amongst differentially expressed genes, while these GO and KO terms are not found to be enriched by Fisher's test, GSEA or PAGE, thus improving the enrichment analysis of microarray data.

6.2.1 Properties of GO and KO terms

First, let us state some properties of gene annotations by GO and KO terms:

- one gene can be annotated by several terms,
- if a gene is annotated by term T then it is annotated by all the ancestors of T, and
- a term may have thousands of genes annotated by it.

From this we can conclude that:

- each GO and KO term defines a gene set,
- one gene can be a member of several gene sets, and
- some gene sets are subsets of other gene sets.

Second, let $Func$ (or $Proc$, $Comp$, respectively) denote the set of gene sets that are defined by the GO terms that are subterms of the term Molecular Function (or Biological Process, Cellular Component, respectively), and let $Path$ denote the set of gene sets defined by the KO terms.

6.2.2 Basic SEGS operators for gene set construction using GO, KO and ENTREZ

Our method relies on two ideas for the construction of new gene sets: using the gene-gene interaction data, and intersection of gene sets.

6.2.2.1 Gene-gene interaction operator

There are cases when some abrupt processes are not detectable by the enrichment score. One of the reasons can be that gene members of that process have a slight increase/decrease in their expression, but this increase/decrease can have a much larger effect on the genes that interact with them. Therefore we propose to construct a gene set whose members interact with members of another gene set (see Figure 6.1). The gene-gene interaction

data can be found in the ENTREZ database. Gene set construction is formally described as follows:

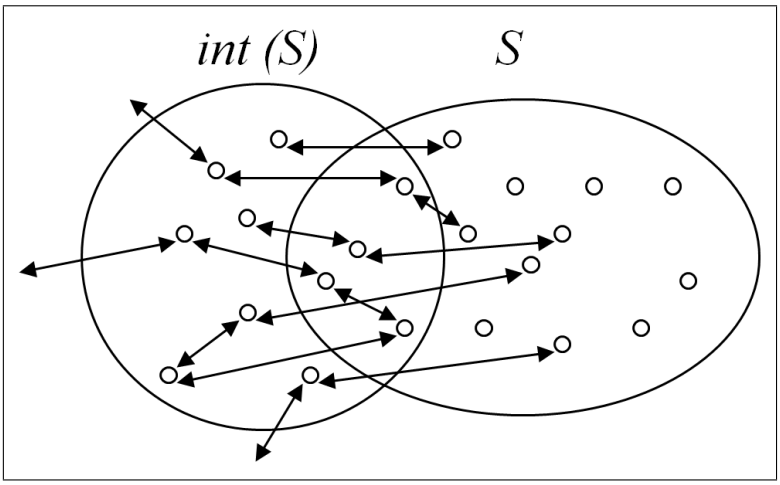


Figure 6.1: Construction of a new gene set, *int(S)*, from existing gene set S. All $g_i \in$ *int(S)* are interacting with some $g_j \in S$. Gene sets S and *int(S)* do not need to intersect.

if $S \in Func$ (or $Proc, Comp, Path$, respectively) then $int(S) = \{g_j \mid g_j$ interacts with $g_i \in S\}$ is added to $Func$ (or $Proc, Comp, Path$).	(6.1)

6.2.2.2 Intersection operator

There are cases where some gene sets are not significantly enriched, but their intersection is significantly enriched. For example, it can happen that a gene set defined by molecular function F is not enriched because a lot of genes in different parts of the cell execute it, and one can not expect that all of them will be over/under expressed, but if genes with that function in a specific part of the cell (C_{part}) are abnormally active, then this can be elegantly described by defining the following gene set:

$$geneSet(S) = func(F),\ comp(C_{part}) = S_F \bigcap S_{C_{part}}.$$

Note that the actual way of constructing new gene sets by intersection of the existing ones is analog to the method of first-order feature construction of the RSD algorithm, described in Section 5.2.1.1. Consequently, viewed as a first-order feature, the gene set S is constructed as a first-order feature:

$$geneSet(S) = func(S, F),\ comp(S, C_{part}).$$

Gene set construction due to gene sets intersection is formally described as follows:

if $S_1 \in Func$, $S_2 \in Proc$, $S_3 \in Comp$ and $S_4 \in Path$, then $S_{new} = S_1 \bigcap S_2 \bigcap S_3 \bigcap S_4$ is a newly defined gene set. (6.2)

An example of this type of construction is presented in Figure 6.2.

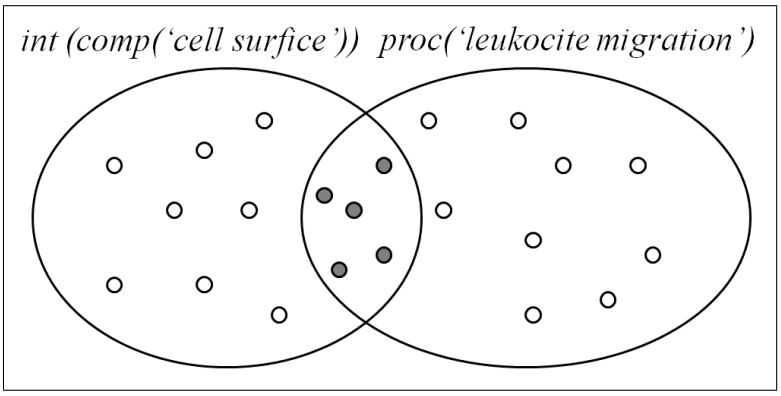


Figure 6.2: Construction of a new gene set, consisting of the members of the 'leukocyte migration' process which interact with genes on the cell surface.

The newly defined gene sets are interpreted very intuitively. For example, gene set S defined as the intersection of 'functional' term A and 'process' term B

$$geneSet(S) = func(A),\ proc(B) \equiv S_A \bigcap S_B$$

is interpreted as: genes that are part of process B and have function A.

The number of potentially newly defined gene sets is huge. It is currently[1] estimated at:

$$|Func| \times |Proc| \times |Comp| \times |Path| \approx 47 \cdot 10^{12} \quad (6.3)$$

If for each of these sets we compute its enrichment score, which in case of GSEA takes linear time in the number of genes ($\approx 2 \times 10^4$), then we need $\approx 10^{18}$ numeric operations. If we want to statistically validate discovered enriched gene sets, usually with 1,000 permutation tests, we get $\approx 10^{21}$ operations, that is well above the average performance of today's PCs. Therefore, we need to efficiently search the gene set space for potentially enriched gene sets, as proposed below.

[1] In September 2007, $|Func| = 7,513$, $|Proc| = 12,549$, $|Comp| = 1,846$ and $|Path| = 272$.

6.2.3 Pruning the search space for enriched gene sets

The first idea for improvement is that we are not interested in generating all possible gene sets, but only those that are potentially enriched. This can be achieved by generating gene sets that have some predefined minimum number of genes at the top of the ranked list, i.e., according to the genes t-scores, for example 3 in the first 100, or 10 in the first 300 genes of the list. That is a weak constraint concerning the biological interpretation of the results, because we are not really interested in gene sets that do not have some minimum number of genes at the top of the list, but it is a hard constraint concerning the pruning of the search space of all gene sets. By having this constraint we can use the GO and KO topology to efficiently generate all gene sets that satisfy the constraint.

As the GO is a directed acyclic graph (DAG), with the root of the graph being the most general term, this means that if one term (gene set) does not satisfy our constraint, than all its descendants will also not satisfy it, because they cover a subset of the genes covered by the given term. In this way we can significantly prune the search space of potentially enriched gene sets. Therefore, we first construct gene sets from the top nodes of the GO and KO, and if we fail to satisfy the given constraint we do not refine the last added term.

Note that the efficiency of the algorithm comes from the usage of the DAG structure of GO and KO. RSD does not use the structure of GO terms when it construct first-order relational features used for describing the genes, but it considers these terms as they have flat structure.

The pseudo code presented in Figure 6.3, implements the basic idea of an efficient construction of potentially enriched gene sets, following the idea of relational feature construction outlined in Section 5.2.1.1.

The main function of the algorithm is the recursive function BUILD-CLAUSE. It tries to add a new term to the given input clause (conjunction of terms). If the new clause covers enough top genes (line 17) then it is added to the resulting list of clauses that describe the new gene sets. After the term is added the procedure recursively calls itself in order to add more terms to the clause (line 21) or to refine the added term (line 25). The provided code will generate all gene sets that have at least 3 genes in the top 100 genes of the GeneList. The proposed method has the data flow model shown in Figure 6.4.

6.3 Experiments

Note that in this study we do not address the problem of discriminating between the classes. Instead, for the given target class we aim at finding relevant enriched gene sets that can capture the underlying biology characteristic for the class.

```
01 topTerm = ['molecular_function', 'biological_process',
02           'cellular_component', 'kegg_pathway']
03
04 function GENERATE-GENE-SETS(GeneList)
05   input: GeneList
06   output: gene_sets
07
08   gene_sets = []
09   BUILD-CLAUSE(0, [], GeneList[1:100], topTerm[0], gene_sets)
10   return gene_sets
11
12 procedure BUILD-CLAUSE(depth, clause, genes, term, gene_sets)
13   input: depth, clause, gene_set, term
14   output: gene_sets
15
16   new_genes = INTERSECTION(genes, TERM_TO_GENES[term])
17   IF LENGTH(new_genes) > 3 THEN             # minimal support ?
18      ADD(clause, term)
19      ADD(gene_sets, clause)
20      IF depth < 4 THEN                       # add more terms
21         BUILD-CLAUSE(depth + 1, clause, new_genes,
22                      topTerm[depth + 1], gene_sets)
23      REMOVE(clause, term)
24      FOR EACH child IN CHILDREN(term) DO     # refine
25         BUILD-CLAUSE(depth, clause, new_genes,
26                      child, gene_sets)
```

Figure 6.3: SEGS algorithm for constructing potentially enriched gene sets.

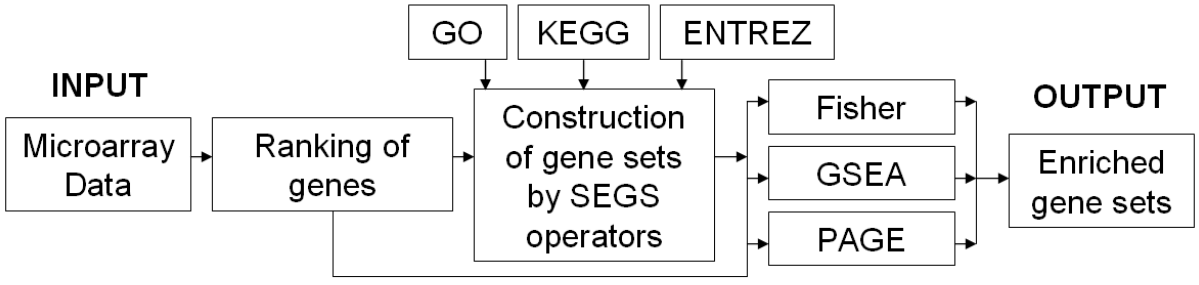


Figure 6.4: Data flow of the proposed SEGS method for the construction of enriched gene sets.

6.3.1 Brief description of datasets

We applied the proposed SEGS methodology to three classification problems: leukemia (32), diffuse large B-cell lymphoma (DLBCL) (71) and prostate tumor (72). All of them are binary classification problems. The leukemia data includes 48 acute lymphoblastic leukemia (ALL) samples and 25 acute myeloid leukemia (AML) samples, each with 7,074 gene expression values. The DLBCL dataset includes 7,070 gene expression profiles for 77 patients, 58 with DLBCL and 19 with follicular lymphoma (FL). The prostate tumor dataset includes 12,533 genes measured for 52 prostate tumor and 50 normal tissue samples. The data for these three datasets were produced from Affymetrix gene chips and are available at http://www.genome.wi.mit.edu/cancer/.

6.3.2 Experimental results

To illustrate the straightforward interpretability of the enriched gene sets found by our approach, we provide the most enriched gene sets for all classes in the three mentioned classification problems (see Tables 6.1, 6.2 and 6.3). Because we use three statistical tests(Fisher's exact test, GSEA and PAGE), which give three different rankings for the enrichment of the gene sets, we calculated the aggregate rank for each gene set by summing its ranks from the separate rankings.

Concerning the number of generated gene sets, for the leukemia dataset we generated 210,762 (ALL) and 127,187 (AML) gene sets, for DLBCL dataset we generated 158,152 (DLBCL) and 78,048 (FL) gene sets, and for the prostate dataset we generated 28,027 (tumor) and 62,567 (normal) gene sets, that satisfied the constraint to have at least 3 genes in the first 100, or 10 in the first 300 most differentially expressed genes. We also set an additional constraint needed for the PAGE algorithm, the size of the generated gene sets, which was chosen to be larger than 30.

6.3.3 Statistical validation

The following procedure was used to calculate the significance of the observed enrichment of a gene set by comparing it with the set of maximal enrichment scores computed from the same datasets but with randomly assigned phenotypes (class labels):

1. Randomly assign the original phenotype (class) labels to samples, reorder genes according to their t-score values, and re-compute the enrichment scores.

2. Repeat step 1 for 1,000 permutations, and create a histogram of the corresponding best enrichment scores for all three tests.

Table 6.1: Five most enriched gene sets (according to the aggregate ranking) found in the leukemia dataset by using GO, KO and ENTREZ. Please note that commas represent the gene sets intersection.

Gene set	Set size	Gene set	Set size
Enriched in ALL		Enriched in AML	
func('DNA binding'), int(comp('nucleoplasm')), int(proc('histone modification'))	41	int(comp('lysosome')), int(proc('response to ext. stimulus')), int(path('Immune System'))	37
int(func('transcrip. repressor activ.')), comp('nucleus'), int(proc('histone modification')), int(path('Long-term potentiation'))	50	int(comp('membrane part')), proc('inflammatory response'), int(path('Human Diseases'))	38
int(func('acetyltransferase activity')), int(comp('nucleus')), int(proc('ubiquitin cycle')), int(path('Signal Transduction'))	45	int(func('peptidase activity')), int(comp('integral to pl. membrane')), proc('defense response')	31
int(func('nucleotidyltransferase activ.')), comp('nucleus'), int(proc('DNA repair')), int(path('Cell cycle'))	84	int(func('metal ion binding')), int(comp('integral to membrane')), proc('inflammatory response')	39
int(func('zinc ion binding')), comp('intracellular organelle part'), int(proc('protein complex assembly')), int(path('Wnt signaling pathway'))	64	int(func('endopept. inhibitor act.')), int(comp('integral to pl. membrane')), int(proc('response to pest.path.par.')), int(path('Cell adhesion molecules'))	43

Table 6.2: Five most enriched gene sets (according to the aggregate ranking) found in the DLBCL dataset by using GO, KO and ENTREZ.

Gene set	Set size	Gene set	Set size
Enriched in DLBCL		Enriched in FL	
int(func('transf.phosph.cont.groups')), int(comp('nuclear part')), proc('biopolymer metabolism')	33	comp('integral to membrane'), proc('humoral immune response')	47
int(func('transf.phosph.cont.groups')), comp('nucleus'), proc('DNA metabolism'), int(path('Cell cycle'))	46	comp('plasma membrane'), path('Hematopoietic cell lineage')	40
int(func('DNA binding')), int(comp('nucleus')), proc('DNA replication'), int(path('Cancers'))	35	func('transmembrane receptor act.'), int(comp('membrane')), int(proc('immune response')), int(path('Immune System'))	83
int(func('DNA binding')), int(comp('nucleus')), proc('biopolymer metabolism'), int(path('Pancreatic cancer'))	50	func('transmembrane receptor act.'), comp('integral to membrane'), int(proc('immune response')), int(path('Env. Inf. Processing'))	100
int(func('transcrip. factor act.')), int(comp('nucleus')), proc('biopolymer metabolism'), int(path('Cell Growth and Death'))	64	proc('humoral immune response'), int(path('Sign. Molec. & Inter.'))	48

Table 6.3: Five most enriched gene sets (according to the aggregate ranking) found in the prostate dataset by using GO, KO and ENTREZ.

Gene set	Set size	Gene set	Set size
Enriched in prostate cancer		Enriched in normal	
func('struct. constituent of ribosome'), comp('intracellular organelle part'), proc('protein biosynthesis'), path('Ribosome')	52	int(func('receptor binding')), comp('integral to membrane'), int(proc('+ regul. of cell prolif.')), int(path('Human Diseases'))	143
func('RNA binding'), comp('ribosome'), proc('protein biosynthesis')	45	int(func('protein kinase act.')), int(comp('integral to membrane')), int(proc('Ras protein sig. transd.')), int(path('Fc eps. RI sig. path.'))	162
func('RNA binding'), comp('cytoplasmic part'), path('Genetic Information Processing')	51	int(func('protein kinase act.')), int(comp('integral to membrane')), int(proc('Ras protein sig. transd.')), int(path('Focal adhesion'))	172
func('struct. constituent of ribosome'), comp('cytost. ribosome (s. Eukaryota)'), proc('protein biosynthesis')	62	int(func('receptor binding')), int(comp('cytosol')), int(proc('+ regul. of cell prolif.')), int(path('Colorectal cancer'))	178
func('RNA binding'), comp('intracellular organelle part')	120	int(func('protein kinase activity')), int(comp('integral to membrane')), int(proc('Ras protein sig. transd.')), int(path('Nat.kill.cell.medi.cyt.'))	170

3. Estimate the p-value for the calculated enrichment score value of the gene set S using the histogram computed at step 2. If there was not a case where random labeling of the examples gives a better enrichment score, then p-value < 0.001.

We use class labeled permutation because it preserves gene-gene correlations and, thus, provides a more biologically reasonable assessment of the significance than the one obtained by randomly permuting the genes.

After the calculation of the gene sets enrichment, we remove gene sets that have too general descriptions. For example, if gene set S_1 is more enriched then gene set S_2, and S_1 has a more specific description than S_2, then S_2 is eliminated. Note that $S_1 = T_{11} \bigcap T_{12} \bigcap T_{13} \bigcap T_{14}$ is more specific than $S_2 = T_{21} \bigcap T_{22} \bigcap T_{23} \bigcap T_{24}$ if T_{1j} is a subterm of T_{2j} for $j = 1 \ldots 4$.

Tables 6.4, 6.5 and 6.6 provide the results of the empirical comparison of SEGS with single GO and KO term analysis of the three datasets. We can see that on all tests the best constructed gene sets are found to be more enriched than the most enriched gene sets defined by taking into account only single GO and KO terms.

Concerning the joint coverage of the five most enriched gene sets, for the ALL class of the first problem, we found that their union consists of 179 genes. The sum of the cardinalities of these five sets is 284. This means that we did not find five different descriptions of the same gene set, but these descriptions cover quite different sets of genes. Similar results were obtained for all the classes of the other two datasets.

Table 6.4: Comparison of the most enriched gene sets constructed using GO, KO and ENTREZ compared to the most enriched gene sets defined by singe GO and KO terms, for the ALL class in the leukemia dataset.

Gene set	Set size	Fisher *p*-value (adj *p*-val)	GSEA *ES* score (adj *p*-val)	PAGE *Z*-score (adj *p*-val)	Aggr. rank (ranks)
Enriched gene sets in ALL (the same as in Table 6.1)					
func('DNA binding'), int(comp('nucleoplasm')), int(proc('histone modification'))	41	$4.18 \cdot 10^{-18}$ (0.001)	0.33 (0.001)	8.92 (0.001)	5 (2+2+1)
int(func('transcrip. repressor activ.')), comp('nucleus'), int(proc('histone modification')), int(path('Long-term potentiation'))	50	$4.96 \cdot 10^{-19}$ (0.001)	0.31 (0.001)	7.37 (0.001)	9 (1+3+5)
int(func('acetyltransferase activity')), int(comp('nucleus')), int(proc('ubiquitin cycle')), int(path('Signal Transduction'))	45	$1.38 \cdot 10^{-17}$ (0.001)	0.21 (0.005)	5.11 (0.015)	16 (3+6+7)
int(func('nucleotidyltransf. activ.')), comp('nucleus'), int(proc('DNA repair')), int(path('Cell cycle'))	84	$1.16 \cdot 10^{-15}$ (0.004)	0.25 (0.002)	5.90 (0.002)	17 (6+5+6)
int(func('zinc ion binding')), comp('intracellular organelle part'), int(proc('protein complex assembly')), int(path('Wnt signaling pathway'))	64	$5.70 \cdot 10^{-16}$ (0.002)	0.28 (0.001)	5.05 (0.021)	19 (5+4+10)
Enriched gene sets in ALL (using single GO and KO terms analysis)					
proc('DNA metabolic process')	314	$9.14 \cdot 10^{-7}$ (0.031)	0.14 (0.018)	4.47 (0.003)	8 (3+4+1)
comp('nucleus')	1461	$3.51 \cdot 10^{-9}$ (0.012)	0.13 (0.020)	3.29 (0.045)	11 (1+5+5)
comp('chromosome')	139	$5.28 \cdot 10^{-7}$ (0.025)	0.19 (0.004)	3.11 (0.061)	15 (2+1+12)
path('pyrimidine metabolism')	48	$9.21 \cdot 10^{-6}$ (0.072)	0.15 (0.010)	4.13 (0.009)	16 (11+3+2)
func('DNA binding')	810	$1.15 \cdot 10^{-6}$ (0.048)	0.10 (0.071)	3.89 (0.011)	18 (7+8+3)
proc('nucleobase, nucleoside, nucleotide & nucleic acid met. proc.')	1321	$4.31 \cdot 10^{-6}$ (0.050)	0.08 (0.125)	3.65 (0.022)	23 (9+10+4)
path('nucleotide metabolism')	101	$1.02 \cdot 10^{-6}$ (0.040)	0.07 (0.144)	3.19 (0.053)	28 (5+13+10)

6.3.4 Biomedical significance of the discovered enriched gene sets

The goal of this study is to provide a better understanding of the biology of malignancies through the use of the background knowledge encoded in GO, KO and ENTREZ. To do so, we have examined biological functions of genes using the entire pathway changes which are more likely (than the changes in the expression of individual genes) to represent meaningful alterations of cellular metabolism in cancers. In its overall design this study fills in the gap of knowledge represented by the common reductionist approach to the interpretation of microarray data whereby increased or decreased expression of a single gene, rather than

Table 6.5: Comparison of the most enriched gene sets constructed using GO, KO and ENTREZ compared to the most enriched gene sets defined by singe GO and KO terms, for the DLBCL class in the lymphome dataset.

Gene set	Set size	Fisher *p*-value (adj *p*-val)	GSEA *ES* score (adj *p*-val)	PAGE *Z*-score (adj *p*-val)	Aggr. rank (ranks)
Enriched gene sets in DLBCL					
int(func('transf.phosph.cont.groups')), int(comp('nuclear part')), proc('biopolymer metabolism')	33	$7.13 \cdot 10^{-16}$ (0.002)	0.36 (0.001)	6.84 (0.001)	3 (1+1+1)
int(func('transf.phosph.cont.groups')), comp('nucleus'), proc('DNA metabolism'), int(path('Cell cycle'))	46	$9.53 \cdot 10^{-16}$ (0.002)	0.29 (0.001)	6.41 (0.001)	6 (2+2+2)
int(func('DNA binding')), int(comp('nucleus')), proc('DNA replication'), int(path('Cancers'))	35	$1.63 \cdot 10^{-15}$ (0.005)	0.24 (0.005)	6.21 (0.001)	11 (3+4+4)
int(func('DNA binding')), int(comp('nucleus')), proc('biopolymer metabolism'), int(path('Pancreatic cancer'))	50	$2.66 \cdot 10^{-15}$ (0.006)	0.26 (0.002)	5.67 (0.007)	12 (4+3+5)
int(func('transcrip. factor act.')), int(comp('nucleus')), proc('biopolymer metabolism'), int(path('Cell Growth and Death'))	64	$4.16 \cdot 10^{-15}$ (0.011)	0.22 (0.008)	6.25 (0.001)	13 (5+5+3)
Enriched gene sets in DLBCL (using single GO and KO terms analysis)					
comp('mitochondrion')	317	$8.23 \cdot 10^{-9}$ (0.018)	0.18 (0.002)	4.92 (0.003)	4 (2+1+1)
proc('DNA replication')	114	$6.72 \cdot 10^{-9}$ (0.015)	0.14 (0.015)	3.97 (0.010)	9 (1+5+3)
func('ATP binding')	567	$7.81 \cdot 10^{-8}$ (0.021)	0.15 (0.004)	3.87 (0.017)	12 (5+2+5)
path('amino acid metabolism')	211	$3.38 \cdot 10^{-7}$ (0.039)	0.15 (0.004)	3.45 (0.046)	18 (6+3+9)
comp('spindle')	29	$8.55 \cdot 10^{-7}$ (0.048)	0.12 (0.025)	3.29 (0.091)	27 (8+7+12)
path('proteasome')	27	$3.14 \cdot 10^{-6}$ (0.053)	0.10 (0.121)	3.73 (0.033)	28 (13+9+6)

behavior of a functionally linked group of genes (a pathway), is used as a readout. In this way, discovered enriched gene sets (described in Tables 6.4, 6.5 and 6.6) for ALL vs. AML, DLBCL vs. follicular lymphoma, and prostate cancer vs. normal tissue, expand our understanding of predictors of clinical behavior of these cancers. Expert interpretation of several found enriched gene sets for each of the three problems is given below.

6.3.4.1 ALL vs. AML

Acute leukemias strike 3-4 people per 100,000 every year. Two major classes of acute leukemias exist: acute lymphoblastic leukemia (ALL) and acute myelogenous leukemia

Table 6.6: Comparison of the most enriched gene sets constructed using GO, KO and ENTREZ compared to the most enriched gene sets defined by singe GO and KO terms, for the TUMOR class in the prostate dataset.

Gene set	Set size	Fisher *p*-value (adj *p*-val)	GSEA *ES* score (adj *p*-val)	PAGE *Z*-score (adj *p*-val)	Aggr. rank (ranks)
Enriched gene sets in TUMOR					
func('struct. constituent of ribosome'), comp('intracellular organelle part'), proc('protein biosynthesis') path('Ribosome')	52	$5.03 \cdot 10^{-17}$ (0.001)	0.40 (0.001)	5.60 (0.001)	5 (3+1+1)
func('RNA binding'), comp('ribosome'), proc('protein biosynthesis')	45	$1.72 \cdot 10^{-18}$ (0.001)	0.32 (0.001)	3.88 (0.019)	9 (1+3+5)
func('RNA binding'), comp('cytoplasmic part'), path('Genetic Information Processing')	51	$7.27 \cdot 10^{-17}$ (0.001)	0.27 (0.003)	4.80 (0.002)	12 (4+6+2)
func('struct. constituent of ribosome'), comp('cytost. ribosome (s. Eukaryota)'), proc('protein biosynthesis')	62	$3.38 \cdot 10^{-16}$ (0.002)	0.31 (0.001)	3.63 (0.024)	20 (7+4+9)
func('RNA binding'), comp('intracellular organelle part')	120	$9.71 \cdot 10^{-16}$ (0.003)	0.23 (0.004)	3.68 (0.021)	27 (12+8+7)
Enriched gene sets in TUMOR (using single GO and KO terms analysis)					
path('Ribosome')	74	$8.64 \cdot 10^{-7}$ (0.028)	0.19 (0.007)	3.39 (0.033)	5 (2+2+1)
comp('cytosolic part')	112	$9.49 \cdot 10^{-6}$ (0.063)	0.21 (0.004)	3.01 (0.058)	11 (5+1+5)
path('Translation')	94	$2.84 \cdot 10^{-7}$ (0.021)	0.12 (0.026)	3.23 (0.050)	14 (1+10+3)
comp('ribonucleoprotein complex')	275	$9.97 \cdot 10^{-6}$ (0.080)	0.16 (0.012)	3.12 (0.054)	16 (7+5+4)
comp('mitochondrion')	462	$8.23 \cdot 10^{-6}$ (0.035)	0.14 (0.021)	2.96 (0.071)	19 (4+8+7)

(AML). The peak incidence of ALL is in childhood (and children account for one quarter of all acute leukemia cases) and it is rare in older adults. In contrast, the median age of AML patients is 60 years and its incidence increases gradually with age. Therefore, as ALL and AML are distinct in clinical presentation, we expected that there would be correlative differences in their biology, as evidenced by microarray expression data.

In fact, the results of our analysis show that functionally linked groups of genes involved in DNA binding (a process whereby transcription factors exert their positive or negative effects on the first phase of protein expression, i.e. transcription of DNA sequence into RNA) and in histone modification (a process whereby transcription machinery is either allowed or prohibited from the access to DNA in the first place) are prominent in ALL cellular pathways, with 41 genes and 50 genes in the first and second ALL gene sets, respectively (22; 26).

This is in agreement with the current understanding of the role of transcriptional activators and repressors in ALL, as is the role of ubiquitin (the third ALL gene set with

45 genes) and DNA repair in this condition (the fourth ALL gene set with 84 genes). Ubiquitin cascade is the major cellular mechanism for recycling proteins, thus regulating their activity and permanence (half-life) in the cell. DNA repair is a key regulator of survival of the cell, normal or malignant, as the unrepaired DNA typically precludes cellular division and proliferation. Lastly, the fifth ALL gene set (64 genes) identifies the evolutionarily conserved Wnt-signaling pathway as active in ALL (85). This is relevant, since Wnt-dependent cellular processes have been shown to be critical for solid organ malignancies, and as therapeutics are already in development for application in solid neoplasms, most notably heaptocellular and colon carcinomas (33; 52), it is plausible that they would have a role in chemotherapy for ALL as well.

Terms identified as relevant in AML include those of immune and inflammatory response, cell adhesion and metal ion binding processes. This perhaps gives extra weights to a recently identified, yet not completely understood, property of AML to be more susceptible to eradication by immune means than ALL (9). In fact, the success of hematopoietic stem cell transplantation for AML maybe in a large part a result of graft-versus-leukemia effect, i.e. immune mediated (68).

6.3.4.2 DLBCL vs. follicular lymphoma

Follicular and diffuse large B-cell lymphomas are two common classes of lymphoma, malignancy that typically involves lymph nodes, spleen, but can originate at other sites, such as gastrointestinal tract, liver, throat, bone, and brain. As expected, immune response pathways (for follicular lymphoma), and DNA binding and replication (key processes in transcriptional regulation of cell division and proliferation in diffuse large B-cell lymphoma) dominate the expression patterns (see Table 6.2) (12; 62).

6.3.4.3 Prostate cancer vs. normal tissue

Prostate cancer is the most common, non-dermatologic male cancer. It represents 33% of cancers and is the third leading cause of cancer deaths in men (84). Thus, the impact on public health is dramatic and any insights with a potential of translation into viable preventive or therapeutic interventions are urgently needed. In this work, the pathways active in gene transcription (upregulated in any rapidly dividing cells, e.g., malignant cell) have been identified: gene sets 1, 2 and 3 in prostate cancer (with 52, 45 and 51 genes, respectively in Table 6.3).

In addition, the investigations of normal cells of prostate point, as expected in normal glandular tissue of prostate, discovered groups of genes involved in cell adhesion, Ras oncogene signal transduction, protein regulation (phosphorylation by kinases), including surface membrane receptors (gene sets 1-5 on normal prostate tissue in Table 6.3).

6.4 Discussion

This chapter addressed the problem of finding enriched functional groups of genes based on gene expression data. The proposed SEGS method allows for integration of GO and KO gene annotations as well as the gene-gene interaction data from ENTREZ into the construction of new interesting relevant gene sets. The experimental results show that the introduced method improves the statistical significance and the functional interpretation of gene expression data, and we base our conclusion on the following facts:

- Enrichment scores of the newly constructed sets are better then the enrichment scores of any single GO and KO term.
- Newly constructed enriched gene sets can be described by non-enriched GO and KO terms, which means that we are extracting additional biological knowledge that can not be found by single term enrichment analysis.
- This method is a generalization of traditional methods. If we turn-off gene-gene interactions and intersections of GO and KO terms, we get the classical single term enrichment analysis.

The results provide strongly suggesting evidence that the proposed SEGS method indeed finds biologically relevant terms not found by single term analysis (see the examples of terms commented by the medical expert in Section 6.3.4). The expert interpretation of the results of this study shows that meaningful analysis of gene products acting jointly in biologically relevant ways is possible and that this and future studies can provide support for transferring of this new technology to clinic.

7 Conclusions and Further Work

In this thesis we first gave an extensive overview of the area of gene expression data analysis, in particular, functional interpretation of gene expression data, we presented an integrated database of different kind of gene information, and presented two new methods for descriptive analysis of gene expression data.

The importance of using biological information as an instrument to understand the biological roles played by genes targeted in functional genomics experiments has been highlighted in this thesis. In recent analysis approaches, genes are no longer the units of interest; the interesting units are groups of genes with a common function. For genes, the available knowledge does not come in the form of sets of clinical variables, but is stored in gene ontology databases where genes are arranged in tree structures according to function, location and other properties. Given a set of genes, one can make a query to a gene ontology database to test if some, say functional, group is over-represented among the genes.

The most common approach is to perform Fisher's exact test to find gene sets annotated by GO terms which are over-represented among the differentially expressed genes. Another way is to apply GSEA or PAGE that uses predefined gene sets and ranks of genes to identify significant biological changes in microarray datasets. However, after correcting for multiple hypotheses testing, few (or no) GO terms may meet the threshold for statistical significance, because the relevant biological differences are small relative to the noise inherent to the microarray technology. In this thesis we present two new methods that approach this problem by expanding the space of gene sets tested for possible enrichment (i.e., checking if the gene set is significantly over-represented in the selected important genes, or if it shows collective over-expression across a list of genes ranked by their differential expression).

The first method uses gene ontology, gene-gene interaction data and the paradigm of relational subgroup discovery to help find patterns of expression for genes with a common biological function that correlate with the underlying biology responsible for class differentiation. The methodology proposes to first select a set of important differentially expressed genes for all classes and then find compact, relational descriptions of subgroups among these genes. We have assessed the quality of the induced descriptions by evaluating them on independent test sets using the cross-validation technique. The results show a clear advantage of using all the complementary sources of background knowledge in the description generation procedure (GO, ENTEZ, as well as degree of differential expression of genes represented by gene weights). We believe that the presented approach can significantly contribute to the application of relational machine learning to gene expression

analysis. Despite the demonstrated benefits of the methodology, the precision and recall evaluation suggests that there is still room for improvement. This is to be achieved through the expected increase in both the quality and quantity of gene/protein annotations in the near future.

The second method address the problem of finding enriched functional gene groups for specific diseases based on gene expression data, GO, KEGG and ENTREZ data. This method is based on the efficient generation of new biologically relevant gene sets, that are tested for possible enrichment. The enrichment of the gene sets was tested with the standard methods: Fisher's exact test, GSEA and PAGE. The new gene sets are generated as intersections of existing GO terms, KEGG terms and gene sets defined with the help of gene-gene interaction data and existing GO and KEGG terms. Our method finds gene sets constructed from GO, KEGG and ENTREZ significantly enriched, and most importantly these single GO and KEGG terms are not found to be enriched by Fisher's test, GSEA or PAGE, thus improving the interpretation of gene set enrichment for microarray data.

A direct comparison between these methods is not possible, because the aim of the first method is to describe the top most differentially expressed genes, and the aim of the second is to find the global biological changes across the whole list of genes, differentially expressed and not differentially expressed genes. Depending of the needs, the user can choose which of the two methods to apply in the analysis.

There are several directions for improvement of the methods. In the next version we plan to extend it with other annotation systems such as gene clusters, chromosomes or common regulatory elements, with which richer biological information might be derived. An extensive study about the relevance of the found enriched gene sets (percentage of false positives) is also planed in the future. Another application of the found enriched gene sets is their usage as features for the classification of microarray data. We believe that some of these features will turn out to be statistically significant markers of specific diseases. At the moment we only provide a web application that implements the second method, but in the future we plan to develop an R package (as R is a standard for implementing and distributing new microarray analysis methods) that will cover both methods and the developed integrated database.

Functional interpretation of gene expression data is still an emerging research area in which a number of issues still need to be addressed. Two main aspects are susceptible of improvement: the definition of blocks of functionally-related genes and the interpretation of data other than simple ranked lists of genes.

Blocks of functionally-related genes refer to biologically meaningful terms that have been defined by curators in different repositories (e.g., GO, KEGG) or can be defined by the users. These blocks can be considered as categorical variables in the sense that a gene belongs (or not) to a given class. Partial or conditional membership is not considered. While this definition could be applicable to some functionally related classes, such as the 'ribosomal proteins' which show a tight coordinated expression, in other classes this level

of coordination in the expression cannot be expected from all the members. Thus, the introduction of weights that consider distinct degrees of membership or the use of different tests that account for non categorical classes would possibly improve the resolution of the methods for functional interpretation of gene expression data.

Not all the experimental outcomes in microarray data analysis can be represented as a list of ordered genes. This representation is suitable for class comparisons or for the study of a continuous parameter (e.g., the level of a metabolite) or survival studies, in which a threshold-free approach can be applied. Nevertheless there are situations in which this list does not have such a simple interpretation, as is the case of multiclass comparisons. Another interesting situation is when multiple phenotype variables are simultaneously studied. In this case instead of a uni-dimensional list the resulting representation could be imagined as a multi-dimensional space in which accumulation of biologically relevant terms must be studied. Also a network of transcriptional interactions could be represented as a graph or as a matrix. In any case, the functional interpretation by threshold-free strategies of these different gene arrangements is something that must be addressed in the future.

Acknowledgements

There are lots of people I wish thank for a huge variety of reasons. Firstly, I wish to thank my supervisor, prof. dr. Nada Lavrač. I could not have imagined having a better adviser and mentor for my PhD, and without her common-sense, knowledge, perceptiveness and suggestions I would never have finished my thesis on time. Her trust in me, letting me choose areas and directions of my work, was the best what I could have got from my adviser.

I wish to thank my thesis evaluators, especially prof. dr. Filip Železný from whom we got the seed idea for this thesis, for managing to read the thesis so thoroughly. I would also like to thank all the rest of the academic and support staff of the Department of Knowledge Technologies at the Jožef Stefan Institute for providing me a comfortable, supportive and stimulating environment.

Much respect to my colleagues, and hopefully still friends, Peter Ljubič, Joël Plisson and Aleksandar Pečkov for putting up with me for more then two years, spending countless hours in discussions connected with our research, but more importantly for things not connected with our work, because without that it would have been much harder to finish this work.

Finally, I have to say 'Thank You' to all my friends and family, wherever they are, for their constant moral support.

References

[1] Al-Shahrour, F., Diaz-Uriarte, R., and Dopazo, J. FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics* (2004), 578–580.

[2] Al-Shahrour, F., and et al. Discovering molecular functions significantly related to phenotypes by combining gene expression data and biological information. *Bioinformatics* (2005), 2988–2993.

[3] Al-Shahrour, F., Minguez, F., and Tarraga, J. BABELOMICS: a systems biology perspective in the functional annotation of genome-scale experiments. *Nucleic Acids Research* (2007), 100–102.

[4] Alexa, A., and et al. Improved scoring of functional groups from gene expression data by decorrelating go graph structure. *Bioinformatics* (2006), 1600–1607.

[5] Alizadeh, A., and et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature* (2000), 503–511.

[6] Ashburner, M., BALL, C., BLAKE, J., and et al. Gene Ontology: tool for the unification of biology. *Nature Genetics* (2000), 25–29.

[7] Ashburner, M., and et al. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research* (2004), 258–261.

[8] Badea, L. Functional discrimination of gene expression patterns in terms of the gene ontology. In *Proceedings of the Pacific Symposium on Biocomputing* (2003), pp. 565–576.

[9] Baron, F., and Storb, R. The immune system as a foundation for immunologic therapy and hematologic malignancies: a historical perspective. *Best Practice & Research Clinical Haematology* (2006), 637–653.

[10] Barry, T., Nobel, A., and Wright, F. Significance analysis of functional categories in gene expression studies: a structured permutation approach. *Bioinformatics* (2005), 1943–1949.

[11] Beissbarth, T., and Speed, T. GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics* (2004), 1464–1465.

[12] Bende, R., and et al. Molecular pathways in follicular lymphoma. *Leukemia* (2007), 18–29.

[13] Benjamini, Y., and Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal Royal Statistical Society* (1995), 289–300.

[14] Benjamini, Y., and Yekutieli, D. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics* (2001), 1165–1188.

[15] Bolstad, B., and et al. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* (2003), 185–193.

[16] Boyle, E., and et al. Go:termfinder-open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics* (2004), 3710–3715.

[17] Brown, M., and et al. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Science USA* (2000), 262–268.

[18] Castillo-Davis, C., and Hartl, D. Genemerge-postgenomic analysis, data mining, and hypothesis testing. *Bioinformatics* (2003), 891–892.

[19] Cestnik, B. Estimating probabilities: A crucial task in machine learning. In *Proceedings of the 9th European Conference on Artificial Intelligence* (1990), pp. 147–149.

[20] Clark, P., and Boswell, R. Rule induction with CN2: Some recent improvements. In *Proceedings of the 5th European Working Session on Learning* (1991), pp. 151–163.

[21] Clark, P., and Niblett, T. The CN2 induction algorithm. *Machine Learning* (1989), 261–283.

[22] Crazzolara, R., and Bernhard, D. Cxcr4 chemokine receptors, histone deacetylase inhibitors and acute lymphoblastic leukemia. *Leukemia & Lymphoma* (2005), 1545–1551.

[23] de Jong, H. Modeling and simulation of genetic regulatory systems: A litterature review. *Journal of computational biology* (2002), 67–103.

[24] Dennis, G., and et al. David: Database for annotation, visualization, and integrated discovery. *Genome Biology* (2003), 3–10.

[25] Draghici, S., Khatri, P., Martins, R., and et al. Global functional profiling of gene expression. *Genomics* (2003), 98–104.

[26] Einsiedel, H., and et al. Histone deacetylase inhibitors have antitumor activity in two nod/scid mouse models of b-cell precursor childhood acute lymphoblastic leukemia. *Leukemia* (2006), 1435–1436.

[27] Eisen, M., Spellman, P., Brown, P., and Botstein, D. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of National Academy of Science USA, 95:25* (1998), pp. 14863–14868.

[28] Fortunel, N., and et al. Comment on 'stemness': 'transcriptional profiling of embryonic and adult stem cells' and 'a stem cell molecular signature'. *Science 302(5644)* (2003), 393–393.

[29] Furey, T., and et al. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* (2000), 906–914.

[30] Ge, H., Walhout, A., and Vidal, M. Integrating 'omic' information: a bridge between genomics and systems biology. *Trends in Genetics* (2003), 551–560.

[31] Goeman, J., Osting, J., Cleton-Jansen, A., and et al. Testing association of a pathway with survival using gene expression data. *Bioinformatics* (2005), 1950–1957.

[32] Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and Lander, E. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* (1999), 531–537.

[33] Gregorieff, A., and Clevers, H. Wnt signaling in the intestinal epithelium: from endoderm to cancer. *Genes Dev* (2005), 877–890.

[34] Grossmann, S., and et al. An improved statistic for detecting over-represented gene ontology annotations in gene sets. *In Proccedings of RECOMB* (2006), 85–98.

[35] Hallikas, O., Palin, K., Sinjushina, N., and et al. Genome-wide prediction of mammalian enhancers based on analysis of transcription-factor binding affinity. *Cell* (2006), 47–59.

[36] Hermjakob, H., and et al. IntAct: an open source molecular interaction database. *Nucleic Acids Research* (2004), 452–455.

[37] Ho, Y., and et al. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature* (2002), 180–183.

[38] Ito, T., and et al. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA.* (2001), 4569–4574.

[39] Jenssen, T.-K., Laegreid, A., Komorowski, J., and Hovig, E. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics* (2000), 21–28.

[40] Johansson, P., and Hakkinen, J. Improving missing value imputation of microarray data by using spot quality weights. *BMC Bioinformatics* (2006).

[41] Kanehisa, M., and et al. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* (2000), 27–30.

[42] Kanehisa, M., Goto, S., Kawashima, S., and et al. The KEGG resource for deciphering the genome. *Nucleic Acids Research* (2004), 227–280.

[43] Karp, P. Pathway Databases: A case study in computational symbolic theories. *Science* (2001), 2040–2044.

[44] Khan, J., and et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* (2001), 673–679.

[45] Khatri, P., and Draghici, S. Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics* (2005), 3587–3595.

[46] Khatri, P., Draghici, S., Ostermeier, G. C., and Krawetz, S. A. Profiling gene expression using onto-express. *Genomics* (2002), 266–270.

[47] Kim, S., and Volsky, D. PAGE: parametric analysis of gene set enrichment. *BMC Bioinformatics* (2005).

[48] Kloesgen, W. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining, AAAI Press* (1996), pp. 249–271.

[49] Kramer, S., Lavrač, N., and Flach, P. Propositionalization approaches to relational data mining. In *Relational Data Mining*, S. Džeroski and N. Lavrač, Eds. Springer, 2001, pp. 262–291.

[50] Lavrač, N., and Flach, P. An extended transformation approach to inductive logic programming. *ACM Transactions on Computational Logic* (2001), 458–494.

[51] Lavrač, N., Železný, F., and Flach, P. RSD: Relational subgroup discovery through first-order feature construction. In *Proceedings of the 12th International Conference on Inductive Logic Programming* (2002), pp. 149–165.

[52] Lee, H., and et al. Wnt/frizzled signaling in hepatocellular carcinoma. *Frontiers in Bioscience* (2006), 1901–1915.

[53] Lee, H., Hsu, A., Sajduk, J., and et al. Coexpression analysis of human genes across many microarray data sets. *Genome Res* (2004), 1085–1094.

[54] Lee, T., and et al. BioWarehouse: a bioinformatics database warehouse toolkit. *BMC Bioinformatics* (2006).

[55] Lewin, A., and Grieve, I. Grouping Gene Ontology terms to improve the assessment of gene set enrichment in microarray data. *BMC Bioinformatics* (2006).

[56] Lockhart, D., and et al. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature biotechnology* (1996), 1675–1680.

[57] Man, M., and et al. POWER_SAGE: comparing statistical tests for sage experiments. *Bioinformatics* (2000), 953–959.

[58] Mao, X., and et al. Automated genome annotation and pathway identification using the KEGG Orthology (KO) as a controlled vocabulary. *Bioinformatics* (2005), 3787–3793.

[59] Mateos, A., Dopazo, J., Jansen, R., and et al. Systematic learning of gene functional classes from dna array expression data by using multilayer perceptrons. *Genome Research* (2002), 1703–1715.

[60] Mewes, H., and et al. MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research* (2004), 41–44.

[61] Oliveros, J., Blaschke, C., Herrero, J., Dopazo, J., and Valencia, A. Expression profiles and biological function. *Genome Informatics* (2000), 106–117.

[62] Paepe, P. D., and Wolf-Peeters, C. D. Diffuse large b-cell lymphoma: a heterogeneous group of non-hodgkin lymphomas comprising several distinct clinicopathological entities. *Leukemia* (2007), 37–43.

[63] Patti, M., and et al. Coordinated reduction of genes of oxidative metabolism in humans with insulin resistance and diabetes: Potential role of pgc1 and nrf1. *Proceedings of the National Academy of Science USA* (2003), 8466–8471.

[64] Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P., Poggio, T., Gerald, W., Loda, M., Lander, E. S., and Golub, T. R. Multiclass cancer diagnosis using tumor gene expression signatures. In *Proceedings of the National Academy of Sciences* (2001), pp. 15149–15154.

[65] Robinson, M., and et al. FunSpec: a web-based cluster interpreter for yeast. *Bioinformatics* (2002), 35–35.

[66] Ross, M. E., Zhou, X., Song, G., Shurtleff, S. A., Girtman, K., Williams, W. K., Liu, H.-C., Mahfouz, R., Raimondi, S. C., Lenny, N., Patel, A., and Downing, J. R. Classification of pediatric acute lymphoblastic leukemia by gene expression profile. *BLOOD* (2003), 2951–2959.

[67] Rual, J., Venkatesan, K., Hao, T., and et al. Towards a proteome-scale map of the human protein-protein interaction network. *Nature* (2005), 1173–1178.

[68] Ruggeri, L., and et al. Natural killer cell alloreactivity in allogeneic hematopoietic transplantation. *Current Opinion in Oncology* (2006), 142–147.

[69] Schaefer, C. Pathway Databases. *Annals of the New York Academy of Sciences* (2004), 77–91.

[70] Schena, M., Shalon, D., Davis, R., and Brown, P. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science* (1995), 467–470.

[71] Shipp, M., and et al. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine* (2001), 68–74.

[72] Singh, D., and et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* (2002), 203–209.

[73] Stelzl, U., Worm, U., Lalowski, M., and et al. A human protein-protein interaction network: a resource for annotating the proteome. *Cell* (2005), 957–968.

[74] Stuard, J., Segal, E., Koller, D., and et al. A gene-coexpression network for global discovery of conserved genetic modules. *Science* (2003), 249–255.

[75] Subramanian, A., and et al. A knowledgebased approach for interpreting genome-wide expression profiles. In *Proc. Natl. Acad. Sci. of the U.S.A.* (2005), pp. 15545–15550.

[76] Subramanian, A., and et al. GSEA-P: A desktop application for gene set enrichment analysis. *Bioinformatics* (2007).

[77] Tavazoie, S., and et al. Systematic determination of genetic network architecture. *Nature Genetics* (1999), 281–285.

[78] Tong, A., and et al. Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science* (2001), 2364–2368.

[79] Tran, T. N., Satou, K., and Ho, T. B. Using inductive logic programming for predicting protein-protein interactions from multiple genomic data. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases* (2005), pp. 321–330.

[80] Uetz, P., and et al. A comprehensive analysis of proteinprotein interactions in saccharomyces cerevisiae. *Nature* (2000), 623–627.

[81] van Noort, V., Snell, B., and Huynen, M. Predicting gene function by conserved co-expression. *Trends in Genetics* (2003), 238–242.

[82] Vencio, R., Koide, T., Gomes, S., and et al. Baygo: Bayesian analysis of ontology term enrichment in microarray data. *BMC Bioinformatics* (2006).

[83] Železný, F., and Lavrač, N. Propositionalization-based relational subgroup discovery with RSD. *Machine Learning* (2006), 33–63.

[84] Walczak, J., and Carducci, M. Prostate cancer: a practical approach to current management of recurrent disease. *Mayo Clinic Proceedings* (2007), 243–249.

[85] Weerkamp, F., and et al. Notch and wnt signaling in t-lymphocyte development and acute lymphoblastic leukemia. *Leukemia* (2006), 1197–1205.

[86] Westfall, P. H., and Young, S. S. Resampling-based multiple testing. *John Wiley & Sons* (1993).

[87] Wrobel, S. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the First European Symposion on Principles of Data Mining and Knowledge Discovery* (1997), pp. 78–87.

[88] Young, A., and et al. Ontologytraverser: an r package for go analysis. *Bioinformatics* (2005), 275–276.

[89] Yuan, Z., Agarwal-Mawal, A., and Paudel, H. K. 14-3-3 binds to and mediates phosphorylation of microtubule-associated tau protein by ser9-phosphorylated glycogen synthase kinase 3beta in the brain. *Journal of Biological Chemistry 279* (2004), 26105–26114.

[90] Zeeberg, B., Feng, W., Wang, G., and et al. Gominer: a resource for biological interpretation of genomic and proteomic data. *Genome Biology* (2003), 28–28.

[91] Zhong, S., and et al. GoSurfer: a graphical interactive tool for comparative analysis of large gene sets in Gene Ontology space. *Applied Bioinformatics* (2004), 261–264.

Appendix 1

The main scientific contributions of this work were published in the following papers:

Conference papers:

- Igor Trajkovski, Filip Železný, Jakub Tolar, Nada Lavrač: Relational Subgroup Discovery for Descriptive Analysis of Microarray Data. *Proceedings of the 2nd International Symposium on Computational Life Science (CompLife)* 2006: LNCS Springer Berlin/Heidelberg, Volume 4216/2006, pp. 86-96, ISBN 978-3-540-45767-1, DOI 10.1007/11875741, Cambridge, UK.
- Igor Trajkovski, Nada Lavrač: Efficient Generation of Biologically Relevant Enriched Gene Sets. *Proceedings of the International Symposium on Bioinformatics Research and Applications (ISBRA)* 2007: LNCS Springer Berlin/Heidelberg, Volume 4463/2007, pp. 248-259, ISBN 978-3-540-72030-0, DOI 10.1007/978-3-540-72031-7, Atlanta, USA.
- Igor Trajkovski, Nada Lavrač: Interpreting Gene Expression Data by Searching for Enriched Gene Sets. *Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME)* 2007: LNCS Springer Berlin/Heidelberg, Volume 4594/2007, pp. 144-148, ISBN 978-3-540-73598-4, DOI 10.1007/978-3-540-73599-1, Amsterdam, The Netherlands.

SCI Journal papers:

- Igor Trajkovski, Filip Železný, Nada Lavrač, Jakub Tolar: Learning Relational Descriptions of Differentially Expressed Gene Groups. *IEEE Transactions on Systems, Man, and Cybernetics, Special issue on Intelligent Computation for Bioinformatics*, Volume 38, No.1, DOI 10.1109/TSMCC.2007.906059, to appear in January 2008.
- Igor Trajkovski, Nada Lavrač, Jakub Tolar: SEGS: Search for Enriched Gene Sets in Microarray Data. *Journal of Biomedical Informatics*, accepted in December 2007 for publication in 2008.

List of Figures

1.1 Data flow of a typical functional interpretation of gene expression data. . 10

2.1 Schematic illustration of cells protein synthesis. The figure is printed by courtesy of the National Human Genome Research Institute, the National Institutes of Health. 18
2.2 Spotted microarray technique. The figure is printed by courtesy of Anna Andersson, Department of Clinical Genetics, Lund University hospital. . . 20
2.3 A part of the GO providing the annotations concerning the positive regulation of muscle cell differentiation. 33
2.4 Histogram of the GO terms on different DAG levels. 34
2.5 Histogram for sizes of GO term annotations. 36
2.6 KEGG pathway 00010 for *glycolysis and gluconeogenesis*. The pathway involves 48 genes in yeast, 55 genes in mouse, and 63 human genes. . . . 37
2.7 This figure shows a part of the KEGG Orthology providing the annotations concerning Carbohydrate and Energy Metabolism. 38

3.1 Threshold-based functional interpretation 43
3.2 Golub data (32), 27 ALL vs. 11 AML samples, 7,074 genes. Left picture is the histogram of the calculated t-scores. Right picture is the histogram of corresponding p-values. There are 1,045 genes with p-value < 0.05. . 47
3.3 Usual scenario of Type I and Type II errors. 48
3.4 Golub data (32), 27 ALL vs. 11 AML samples, 7,074 genes. 681 genes are selected as differentially expressed. 49
3.5 Threshold-free functional interpretation 53
3.6 The 'spectral lines' show the positions of genes members of gene set S on the ranked gene list. This figure is borrowed from the supplementary material of (75). 54
3.7 Histograms of t-score values from Golub dataset (32), when one gene is selected (left) and histogram of the average of t-scores of 10 randomly selected genes (right). 55

4.1 Part of GO-KO ontology. In September 2007, the GO-KO ontology has about 22,000 terms, of which 273 are KO terms. 61
4.2 A part of data, providing annotation of the gene LDHA lactate dehydrogenase with KEGG and GO terms, contained in the ENTREZ database. . 62
4.3 Part of gene annotation database. 63
4.4 Part of gene-gene interactions database. In September 2007, the number of all gene-gene interactions was about 118,000. 64

5.1 Data flow of the method for learning relational descriptions of DE genes . 69
5.2 Descriptions of discovered subgroups ideally cover just individuals of the target class (subgroups 1 and 3), however they may cover also a few individuals of other classes (subgroup 2). 74
5.3 Expression of three genes (A, B and C) for five patients of class 1 and five patients of class 2. Perfect class distinction can be achieved by idealized gene A, in which the expression level is uniformly low in class 1 and uniformly high in class 2. A more realistic case is gene B which is also useful for class distinction. We do not use gene C for class distinction as we are interested in genes that have significant difference in their mean expression between the classes. 78

6.1 Construction of a new gene set using interactions 91
6.2 Construction of a new gene set by intersection 92
6.3 SEGS algorithm for constructing potentially enriched gene sets. 94
6.4 Data flow of the proposed SEGS method for the construction of enriched gene sets. 94

List of Tables

3.1 GO terms found to be differentially distributed when comparing ten independent random partitions of 50 genes sampled from the complete genome of yeast. 45

3.2 Compilation of tools for functional interpretation of gene expression data. Although the most common tools have been included here, this list is not exhaustive. 50

5.1 Average, maximal and minimal value of $|T(g, c)|$, for $g \in G_C(c)$, for each problem and class c. 79

5.2 Precision, recall and AUC figures of found subgroups, for the set of ALL/AML, Subtypes of ALL and Multi-Class-Cancer differentially expressed genes, obtained through 5-fold cross-validation. 82

5.3 Precision of discovered differentially expressed gene group descriptions, for three scenarios where part of the background knowledge or gene-weight information was removed. 83

6.1 Five most enriched gene sets (according to the aggregate ranking) found in the leukemia dataset by using GO, KO and ENTREZ. Please note that commas represent the gene sets intersection. 96

6.2 Five most enriched gene sets (according to the aggregate ranking) found in the DLBCL dataset by using GO, KO and ENTREZ. 96

6.3 Five most enriched gene sets (according to the aggregate ranking) found in the prostate dataset by using GO, KO and ENTREZ. 97

6.4 Comparison of the most enriched gene sets constructed using GO, KO and ENTREZ compared to the most enriched gene sets defined by singe GO and KO terms, for the ALL class in the leukemia dataset. 98

6.5 Comparison of the most enriched gene sets constructed using GO, KO and ENTREZ compared to the most enriched gene sets defined by singe GO and KO terms, for the DLBCL class in the lymphome dataset. 99

6.6 Comparison of the most enriched gene sets constructed using GO, KO and ENTREZ compared to the most enriched gene sets defined by singe GO and KO terms, for the TUMOR class in the prostate dataset. 100

Extended abstract

Microarrays are at the center of a revolution in biotechnology, allowing researchers to simultaneously monitor the expression of tens of thousands of genes. The final aim of a typical microarray experiment is to find a molecular explanation for a given macroscopic observation (e.g., which pathways are affected by the loss of glucose in a cell, what biological processes differentiate a healthy control from a diseased case); this is called *functional interpretation* of gene expression data.

Introduction

First methods for functional interpretation of microarray data used a two-step approach, in which first genes of interest are selected. Typical criteria for selection are differential expression or co-expression. Then in the second, independent step, the annotations of these genes by biologically functional terms are analyzed, usually by looking for functional terms over-represented in the group of genes selected in the first step. Examples of widely used terms with functional meaning are *Gene Ontology* (GO) and *Kyoto Encyclopedia of Genes and Genomes* (KEGG) pathways. Programmes such as OntoExpress, FatiGO, GOMiner, etc., can be considered as representatives of a family of methods that use these terms to find clues for the interpretation of the results of microarray experiments. By means of this simple two-step approach, a reasonable biological functional interpretation of a microarray experiment can be attained.

Nevertheless, this approach has a weak point: the resulting list of genes of interest is generally incomplete. This is due to the fact that the definition of this list is affected by many factors including, among others, the method of selection and the imposed thresholds during the analysis. That is one of the reasons that initiated the development of a new generation of procedures which draw inspiration from molecular systems biology. These procedures aim to directly test the behavior of blocks of functionally related genes, instead of focusing only on the most differentially expressed genes. The *Gene Set Enrichment Analysis* (GSEA) *and Parametric Analysis of Gene Set Enrichment* (PAGE) have pioneered a family of methods devised not to find individual genes but to search for groups of functionally related genes with a joint (although not necessarily high) over- or under-expression across a list of genes ranked by their differential expression between classes of microarray data.

Even with the introduction of new methods, very often after correcting for multiple hypotheses testing, few (or no) GO or KEGG terms turn out to meet the threshold for statistical significance, because the relevant biological differences are small relative to the noise inherent to the microarray technology.

Scientific contribution

This thesis presents two new methods for the functional interpretation of gene expression data that combine and use knowledge stored in different kinds of biological databases. The interpretation is done by identifying and describing gene sets that have significantly altered expression profile (e.g., over- or under-expressed). The search of the interesting gene sets is performed in the space of already defined gene sets (genes that have common annotation by predefined ontological terms) and in the space of newly generated gene sets that have predefined characteristics (e.g., the minimum number of member genes that are found to be differentially expressed). Three well established methods, Fisher's exact test, GSEA, and PAGE, were employed in order to identify gene sets with significantly altered expression profiles.

Both developed methods share the same mechanism of first-order (relational) feature construction, by using the GO, KEGG Orthology, gene annotations, and gene-gene interaction data. These features, constructed by the propositionalization mechanism of the Relational Subgroup Discovery algorithm (RSD), are used as generalized gene annotations.

Learning Relational Descriptions of Differentially Expressed Gene Sets

This method belongs to the class of threshold-based functional analysis methods. It is performed in two steps. In the first step, 'top' genes of interest are *selected* using gene differential expression as a selection criterion. The selection process does not take into account the fact that gene products are acting cooperatively in the cell and consequently, for better interpretation of the selected gene list, in the second step their behavior must be coupled to some extent by looking for their common description. The language used for describing the functionality of the genes is constructed from GO, gene annotations, and gene-gene interaction data. By using this background knowledge together with the paradigm of relational subgroup discovery we found common descriptions of gene sets differentially expressed in specific cancers. The

descriptions of these gene sets can be straightforwardly used by the medical experts.

The input to our algorithm is a multi-dimensional numerical dataset, representing the expression of the genes under different conditions (that define the classes of examples), GO, and gene-gene interaction data used for producing background knowledge about these genes. The output is a set of gene sets whose expression is significantly different for one class compared to the other classes.

It is noteworthy that the latter descriptive 'post-processing' step is a machine learning task, in which the curse of dimensionality usually ascribed to microarray data classification, actually turns into an advantage. This is because, in traditional microarray data mining configurations, the high number of genes results in a high number of attributes usually confronted with a relatively small number of expression samples, thus forming grounds for overfitting. In our approach, on the contrary, genes correspond to examples and thus their abundance is beneficial. Furthermore, the dimensionality of the secondary attributes (relational features of genes extracted from gene annotations) can be conveniently controlled via suitable constraints of the language grammar used for the automatic construction of the gene features.

We apply the proposed methodology on three classification problems from gene expression data (ALL vs AML leukemia, subtypes of ALL leukemia and classification of 14 types of cancers), with the aim to describe the genes that are usually used by the classifiers, i.e, the differentially expressed genes. The results provide strongly suggesting evidence that the proposed method indeed finds biologically relevant terms not found by single term analysis. The expert interpretation of the results of this study shows that meaningful analysis of gene products acting jointly in biologically relevant ways is possible and that this and future studies can provide support for transferring of this new technology to clinic.

We believe that the developed method can significantly contribute to the application of relational machine learning to gene expression analysis especially through the expected increase in both the quality and quantity of gene/protein annotations in the near future.

SEGS: Search for Enriched Gene Sets

This is based on threshold-free functional analysis. This method is also performed in two steps. In the first step, genes are *ranked* by using their differential expression values when comparing predefined classes (e.g., tumor vs. healthy controls) by means of a appropriate statistical test (e.g., the t-test). In the second step, the positions of the members of the predefined gene sets (e.g., defined by GO and KEGG Orthology (KO) terms) in the ranked list are analyzed using appropriate statistical tests (e.g., the Kolmogorov-Smirnov test). Gene sets, whose members are predominantly found at the top of the list, are considered enriched and responsible for the phenotype difference (e.g., the tumor vs. normal). Our contribution to this methodology is a development of an *efficient algorithm*, inspired by the RSD first-order features construction, for the construction of *new*, potentially enriched, gene sets. New gene sets are defined by conjunctions of relational features constructed from the background knowledge.

Testing the enrichment of these gene sets with the standard methods (Fisher's exact test, GSEA and PAGE) shows that our method finds gene sets constructed from GO and KO terms significantly over-represented amongst differentially expressed genes, while these GO and KO terms are not found to be enriched by Fisher's test, GSEA or PAGE, thus improving the enrichment analysis of microarray data.

We applied the proposed SEGS methodology to three classification problem datasets: leukemia, diffuse large B-cell lymphoma and prostate tumor, with aim to find relevant enriched gene sets that can capture the underlying biology characteristic for the given class. The experimental results show that the introduced method improves the statistical significance and the functional interpretation of gene expression data, and we base our conclusion on the following facts: Enrichment scores of the newly constructed sets are better then the enrichment scores of any single GO and KO term, and newly constructed enriched gene sets can be described by non-enriched GO and KO terms, which means that we are extracting additional biological knowledge that can not be found by single term enrichment analysis.

Construction of an integrated database

Different kinds of information and data are spread over the web, hosted in a large-scale independent, heterogeneous and highly focused resources. While

the time to obtain genomic data is getting shorter, the time for one to process the data and understand the biological meaning is much prolonged. Therefore, the integration of biological data and information has become an important ongoing scientific problem, as researchers have not yet been offered comprehensive tools for integrative data and information processing.

We approach this problem by dividing it in three parts: creation of a database for genomic data and information, creation of a platform for analyzing the gene expression data, and creation of a web-based tool for accessing the data and knowledge discovery. We created an easy to use relational database that integrates numerous public databases (GO, KO, gene annotations and gene-gene interaction data) in a common, structured format, placing a broad and deep set of searchable information at the fingertips of researchers of the wider scientific community.

Conclusions and further Work

The two developed methods have proved to be of interest to medical experts. The extracted knowledge turns out to be consistent with the relevant literature, and proves to have the potential for guiding the biomedical research and generating new hypotheses that explain microarray measurements.

Direct comparison between these methods is not possible, because the aim of the first method is to describe the top most differentially expressed genes, and the aim of the second is to find the global biological changes across the whole list of genes, differentially expressed and not differentially expressed genes. Depending of the needs, the user can choose which of the two methods to apply in the analysis.

As a further work, an extensive study about the relevance of the found enriched gene sets (percentage of false positives) is planed in the future. Next further work will also aims at using discovered enriched gene sets as features for classification of microarray data. We believe that some of these features will turn out to be statistically significant markers of specific diseases.

Biography

Igor Trajkovski was born in Skopje, Macedonia, on 12 June 1977.

He completed his Bachelor of Science degree in computer science at the Ss. Cyril and Methodius University - Skopje, Faculty of Natural Sciences and Mathematics, Skopje, Macedonia, in 2001. After completing his undergraduate study he completed his Master of Science degree in computer science, with specialization in bioinformatics, at Saarland University and Max Planck Institute for Informatics, Saarbruecken, Germany, in 2004.

During his studies he successfully participated in several international programming competitions, including International Olympiad in Informatics in 1996, Balkan Olympiad in Informatics in 1996 (winner of the Bronze medal) and the World Final ACM collegiate programming contest in 2000. His research interests include machine learning, bioinformatics, gene expression data analysis, fundamental algorithms and data structures, parallel algorithms and common sense representation and reasoning.