The 23rd ECMI Modelling Week
European Student Workshop
on Mathematical Modelling
in Industry and Commerce

Wrocław University of Technology, Wrocław, Poland, August 23-30, 2009

Report of project group 3 on

How to enhance the exploratory power of Qlucore Omics Explorer?

Project 3: How to enhance the Exploratory Power of the Qlucore Omics Explorer

Supervisor:

Magnus Fontes (Lunds Universitet, Sweden)

Group members:

Sylvestre Burgos (University of Oxford, United Kingdom)

Manh Hong Duong (ESIM, TU Kaiserslautern, Germany)

Laura Friis Frølich (DTU Lyngby, Denmark)

Daniel Høyer Iversen (NTNU Trondheim, Norway)

Bogna Pawłowska (Politechnika Wrocławska, Poland)

Matthias Voigt (TU Chemnitz, Germany)

Mikalai Zhudro (ESIM, TU Eindhoven, The Netherlands)

November 1, 2009

Contents

Abstract

Qlucore Omics Explorer (QOE) is a data analysis tool from Qlucore. Using QOE, a researcher can explore huge datasets containing genetic information to look for patterns and structure. Powerful statistical methods in QOE secure that possible findings are statistically relevant. In this article we show how to enhance the exploratory power of QOE by additionally considering lists of genes with known biological functions from databases.

For this purpose we compare lists we get from QOE with those provided by the databases. We show a way to estimate the probability that these two have a certain number of common elements. First we assume that all genes contained in the lists have the same probability. Later we drop this assumption and develop a model for lists where the genes may have different probabilities to appear. To do so we group the genes in several probability classes and use a multivariate hypergeometric distribution. In this way we develop recommendations for future improvements of QOE.

Introduction

Recently, thanks to new experimental methods such as the microarray technology it is possible to analyse the whole human genome. Thanks to some companies offering appropriate software and hardware, scientists were able to discover and describe functions of thousands of sets of genes which are saved in public databases. Nowadays all these databases can be found on internet. The question is how to use them and how to apply all this knowledge practically. Qlucore Omics Explorer is a program which can deal with the data provided from the databases and analyse it statistically. In the paper we show how to use the databases in QOE and describe a method to get more useful biological information.

Chapter 1

Gene Databases and Qlucore Omics Explorer

This chapter gives a short introduction to the biological databases on genes and Qlucore Omics Explorer (QOE). We also describe how to use QOE to create a list of genes.

1.1 Biological Databases

Scientists have already done a lot of experiments on genes. Their findings are placed in some public repositories and databases. There we can find information about DNA, the genome, gene sequences, experimental data and results. We can divide these repositories into 2 broad categories:

- Databases with public data of experiments
- Databases with collections of gene sets

A list of all existing databases (in 2009) can be found at
http://www3.oup.co.uk/nar/database/cat/9.

Next, we will briefly discuss these databases, their usefulness and their use in Qlucore Omics Explorer.

1.2 Databases with Collection of Gene Sets (Microarray Data and Gene Expression Databases)

In this section we briefly introduce some databases.

Gene Expression Omnibus
GEO is a biological database maintained by the National Center for Biotechnology Information (NCBI). It contains information of experiments measuring the abundance of mRNA, miRNA, genomic DNA and proteins in dual-channnel microarray format. All the data supports MIAME compliant data submission. Information about the MIAME format can be found at
http://www.ncbi.nlm.nih.gov/geo/info/MIAME.html
You can query data sets, gene profiles, GEO accession and GEO Blast. Besides you can browse data sets and GEO accessions. Information about the experiments is mostly provided in Affymetrix CEL format. For using this data in Qlucore Omnics Explorer, the tool for the transformation between the CEL and Chip file format can be useful.

Array Express
Since its creation, this database's main goals were to support publication and to provide high quality information about gene expression. As for the previous database, it has a big amount of gene expression experiments in MIAME format.

Gene Ontology
This database provides common controlled vocabulary for annotating genes, gene sequence and products. It is also thel main source for gene ontologies. Gene ontology is a collection of gene sets structured in functional categories based by known or predicted behavior. These functional categories are biological processes, molecular functions and cellular components. Now the database contains more than 20.000 datasets.

MSigDB
This is a gene set collection that is to be used with the GSEA (Gene Set Enrichment Analysis) software. You can search, browse, download, annotate gene sets, and view annotations. When you provide your own gene sets, you can compute overlaps between your findings and the existing knowledge. This project also provides software for working with the GSEA method.

Finally we give a short list of useful links.

1. Information about GEO in the journal "Nuclear Acids Reserch" - http://www3.oup.co.uk/nar/database/summary/603
2. GEO - http://www.ncbi.nlm.nih.gov/geo/
3. List of databases - http://www3.oup.co.uk/nar/database/cat/9
4. Array Express - http://www.ebi.ac.uk/arrayexpress
5. Information about Array Express in "Nuclear Acids Reserch" - http://www3.oup.co.uk/nar/database/summary/338
6. MIAME format description - http://www.ncbi.nlm.nih.gov/geo/info/MIAME.html
7. Gene Ontology - http://www.geneontology.org/
8. Molecular Signature Database - http://www.broadinstitute.org/gsea/msigdb/index.jsp

1.3 Introduction to Qlucore Omics Explorer

QOE is a product of the company Qlucore. The software is used in data analysis and data mining and is built on mathematical and statistical methods. Using QOE one can:

- find patterns and structures in large data sets;
- find variable dependances;
- work with synchronized heat map and PCA plots.

QOE can work with the following types of data:

- gene expression: microarrays, real-time PCR;
- microRNA: microarrays, real-time PCR;
- DNA methylation: microarrays;
- protein expression: microarrays, antibody arrays, 2-D gels;

- image analysis data;
- any data set of multivariate data of sizes up to 1000 samples and 100,000 variables or 1000 variables and 100 000 samples.

The data to be imported must be in the following formats:

- Qlucore Data Files (*.gedata);
- BioArray Software Environment Files (*.base);
- Affymetrix Probe Set Files (*.chp).

More information about QOE can be found on the website of the company Qlucore [8].

1.4 How to use QOE to create a List of Genes

In our project we use QOE to create a list of genes. We then compare it to the list obtained from medical doctors and give a method to hopefully extract some useful information from the overlap. In this subsection will introduce how to create a list of genes using QOE. We will illustrate this by using the example data available in QOE: the Acute Lymphoblastic Leukemia.gedata

Step 1. Loading the Data
After installing QOE and opening it, we can load the example data by choosing:

Help > Example Files > Acute Lymphoblastic Leukemia.gedata

in the Menu bar.

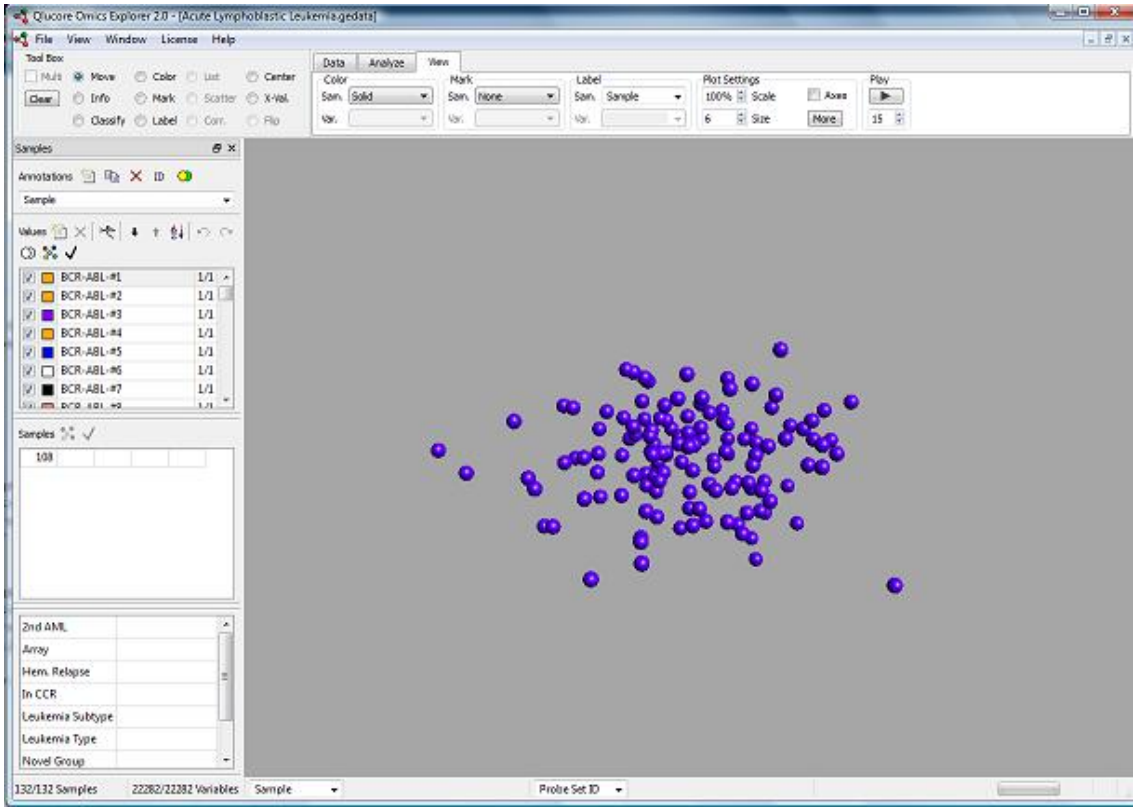


Figure 1.1: The data is loaded.

When the data is loaded we see 132 samples with 22282 variables in the toolbar.

Step 2. Working with Samples
In the annotation box, select **Leukemia Subtype** from the list of **Sample Annotations**

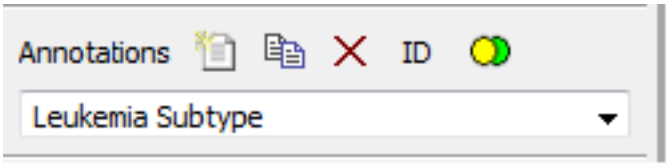


Figure 1.2: Annotations

and click on the **Sample Colors** button  to color the data.

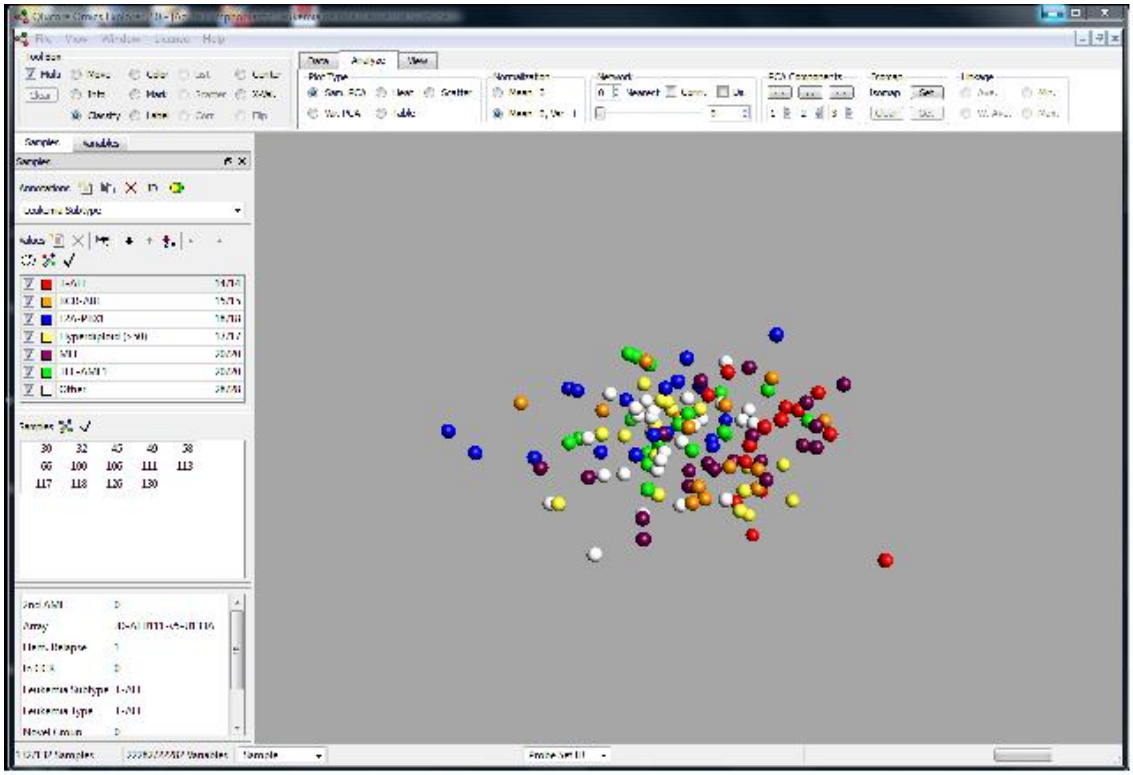


Figure 1.3: The colored data.

However, it is difficult to recognize any structure or pattern in the this data. The next step consists of using statistical analysis to discern pattern in the data sets.

Step 3. Using statistical Analysis
Open the statistics dock window (**View** > **Dock Window** > **Statistics**) and set the value on the **Filter by Variance** slider to 0.3.

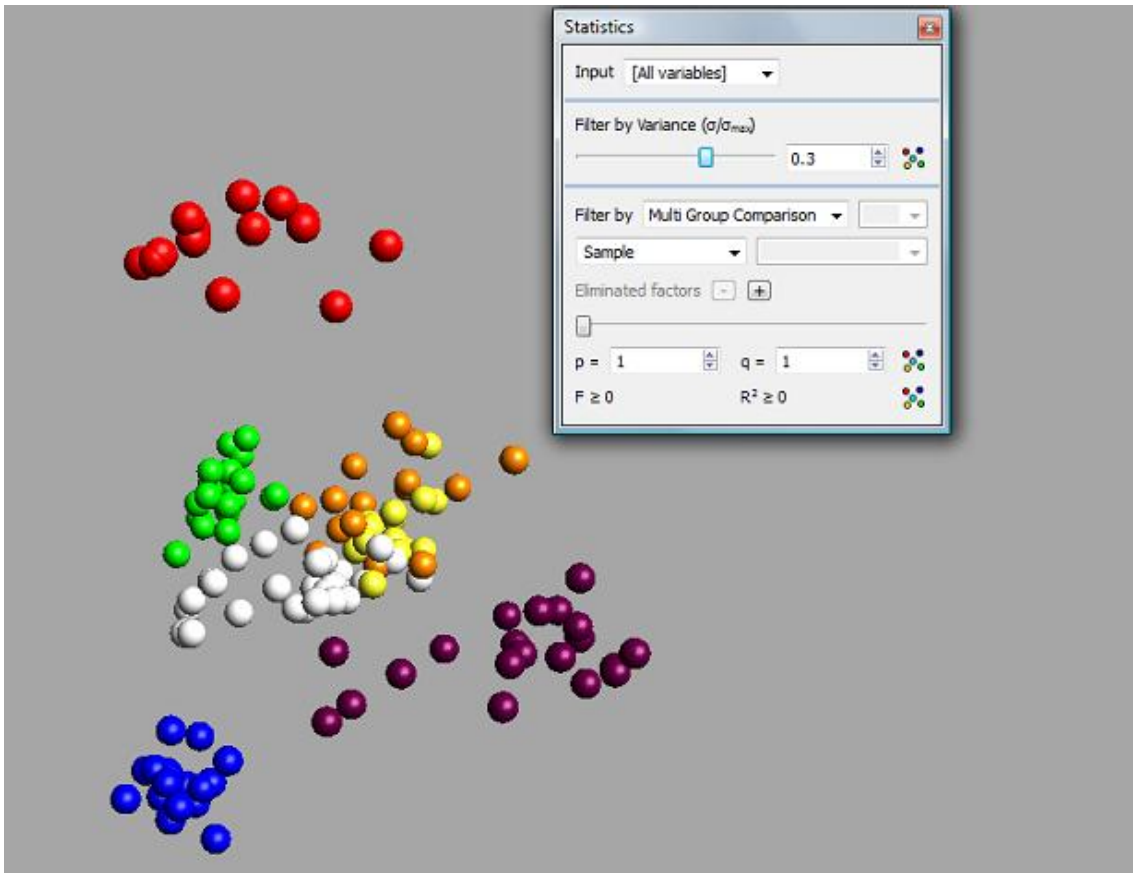


Figure 1.4: The grouped samples after statistical analysis

By using the principal component analysis we see that the T-ALL subgroup (the red one in the plot) clearly distinguishes itself from the rest of the subtypes. Therefore, we can eliminate the corresponding samples and analyse the other groups. To remove the T-ALL group just uncheck it. Then we can consider the other groups more precisely.

- Select **Multi Group Comparison** in the **Statistics** window.
- Select **Leukemia Subtype** in the corresponding Combo box.
- Set up the p to $1e-7$.
- In the **Network** tab, set the number of nearest neighbours to 7.

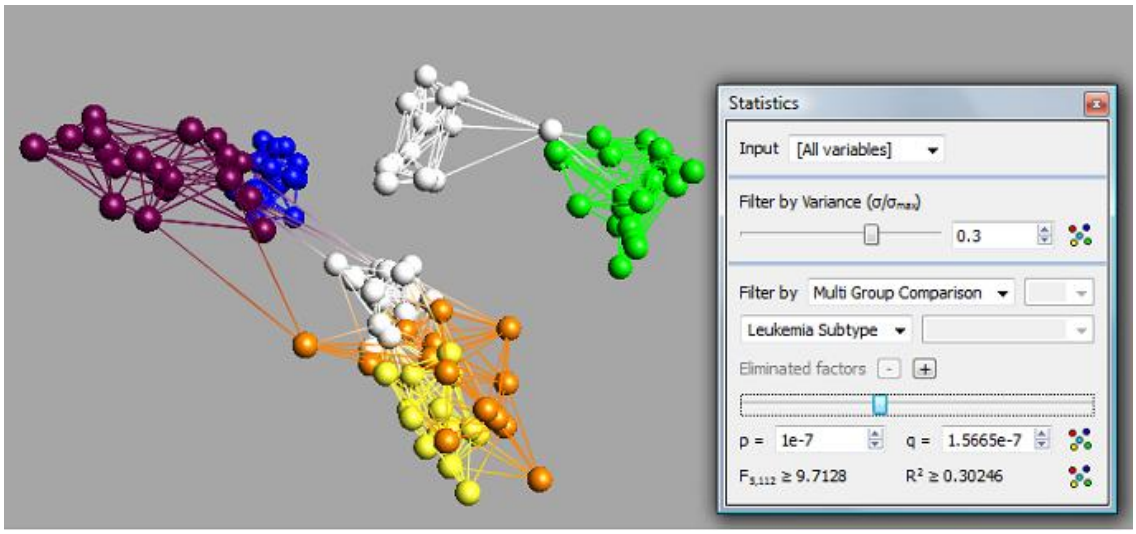


Figure 1.5: The network of samples with their nearest neighbours

In the plot above we see that the white group contains two distinct subgroups. These subgroups do not share any of their 7 nearest neighbors. Therefore they should not be put in the same class and we should redefine the white group as two distinct subgroups.

Step 4. Modifying the Groups

- Select **Classify** in the **Toolbox** window.
- Select the **New Value** button in the **Sample Value** panel in the **Sample** dock window. A new value will appear in the **Value** table.

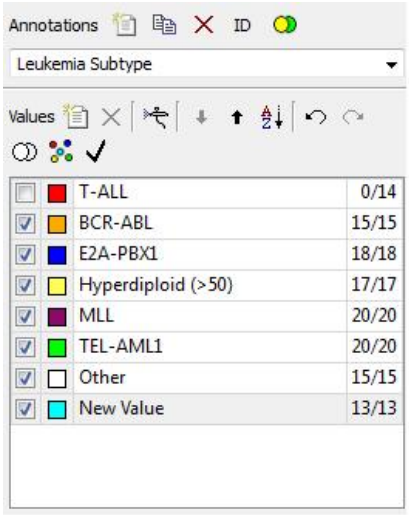


Figure 1.6: The Sample window

- Select the subgroup of the white group which is closest to the green group.

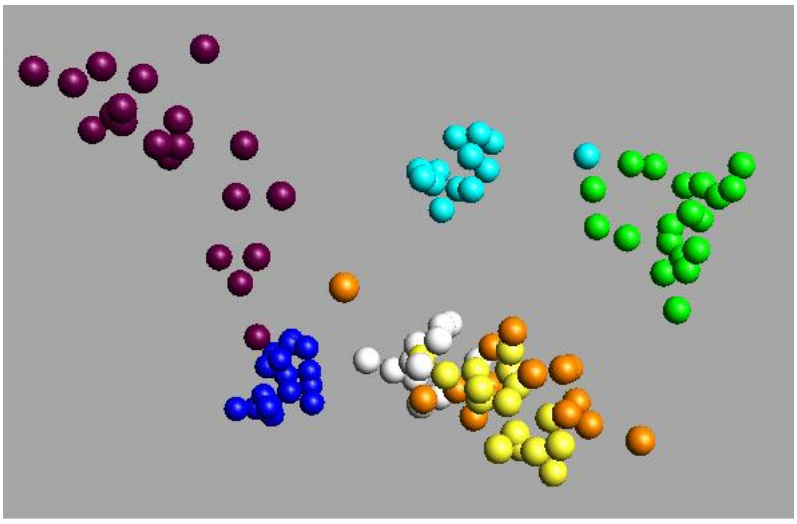


Figure 1.7: The modified groups

Step 5. Finding Variables that discriminate two Groups
In the previous step we knew that the white group consisted of two different subgroups and we have redefined them. In this step we will find the variables that discriminate two groups and create a corresponding gene list.

- Select **Window** > **New Synchronized Plot** in the **Menu** bar.
- Select **Window** > **Tile** in the **Menu** bar.
- Select **Novel Group** in the **Sample Annotation** textbox in the **Sample** dock window.
- Select the **Sample Color** button in the **Sample Annotations** toolbar.

We will get the following plot.

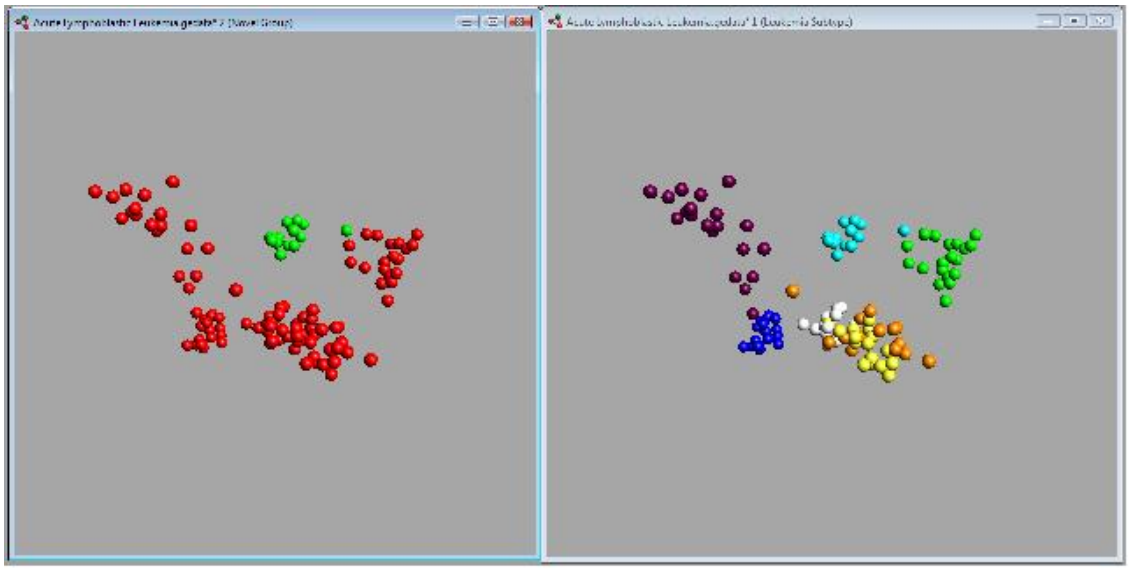


Figure 1.8: The novel group

- Select **Var. PCA** in the **Analyze** tab in order to display a variable PCA plot.

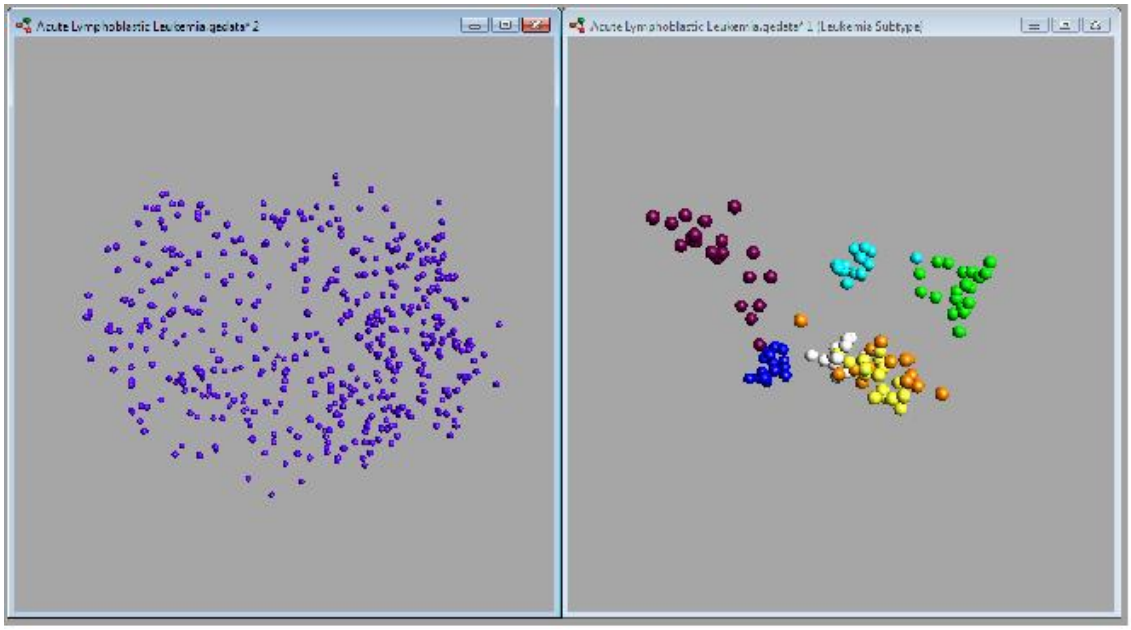


Figure 1.9: The variable PCA plot

- Select **New Value** in the **Value** table.
- Select the **Variable Color** button in the **Value** toolbar.
- Select the **Variables** dock window.
- Select the **New** button in the **Variable Lists** toolbar.
- Select **List** in the **Toolbox**.
- Draw a closed clockwise curve around some of the genes that are red.

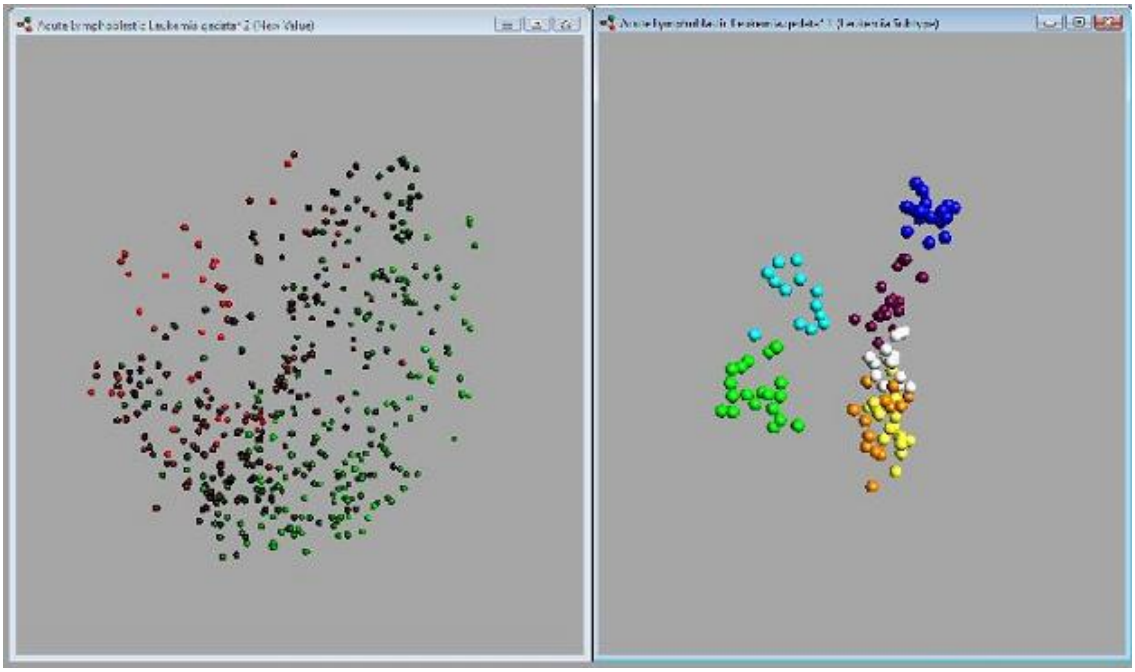
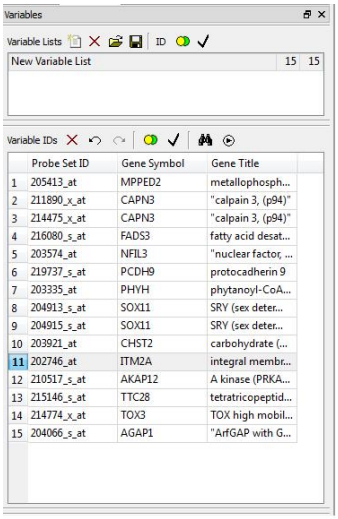


Figure 1.10: Variables with corresponding gene expression (red - high expression, green - low expression)

We will then create a list of genes that have the highest expression in the new group. The variables' names list appears in the **Variable** table displaying the selected genes.



	Probe Set ID	Gene Symbol	Gene Title
1	205413_at	MPPED2	metallophosph...
2	211890_x_at	CAPN3	"calpain 3, (p94)"
3	214475_x_at	CAPN3	"calpain 3, (p94)"
4	216080_s_at	FADS3	fatty acid desat...
5	203574_at	NFIL3	"nuclear factor, ...
6	219737_s_at	PCDH9	protocadherin 9
7	203335_at	PHYH	phytanoyl-CoA...
8	204913_s_at	SOX11	SRY (sex deter...
9	204915_s_at	SOX11	SRY (sex deter...
10	203921_at	CHST2	carbohydrate (...
11	202746_at	ITM2A	integral membr...
12	210517_s_at	AKAP12	A kinase (PRKA...
13	215146_s_at	TTC28	tetratricopeptid...
14	214774_x_at	TOX3	TOX high mobil...
15	204066_s_at	AGAP1	"ArfGAP with G...

Figure 1.11: A gene list containing genes with the highest expression

Chapter 2

Gene Set Enrichment Analysis - An Overview

In this section we give a short introduction to the methodology of the Gene Set Enrichment Analysis (GSEA). For a full mathematical description we refer the reader to [5] and to [2] to read about recent improvements of the method.
Assume that we have n samples of genome-wide expression profiles that can be determined by evaluating microarray experiments. Assume additionally that the set of samples may be divided into two groups: the treatment group contains n_1 samples that represent a characteristic trait, e.g. the individuals whose samples belong to this group suffer from a specific disease. The complement of size $n_2 = n - n_1$ samples is called the control group and thus represents individuals without that characteristic trait. Furthermore let S be a given set of genes that represents a known biological function. The goal of the GSEA is simply to check whether the two groups (phenotypes) differ significantly in their gene expressions on the set S (or several gene sets). We will now describe how the GSEA method works.

2.1 Calculation of a Ranked Gene List L

. First of all it is necessary to rank all genes of the genome in a list L such that the genes that have the highest expressions in the treatment group compared to the control group appear at the beginning of the list L (upregulated genes) and that genes with the lowest expressions in the treatment group compared to the control group appear at the end of the list L (downregulated genes), respectively. This difference in the gene expressions may be computed by a two-sample t-statistic $t(g_i) = t_i$ for every of the N genes g_i of the genome (see any statistics book, e.g. [4]).

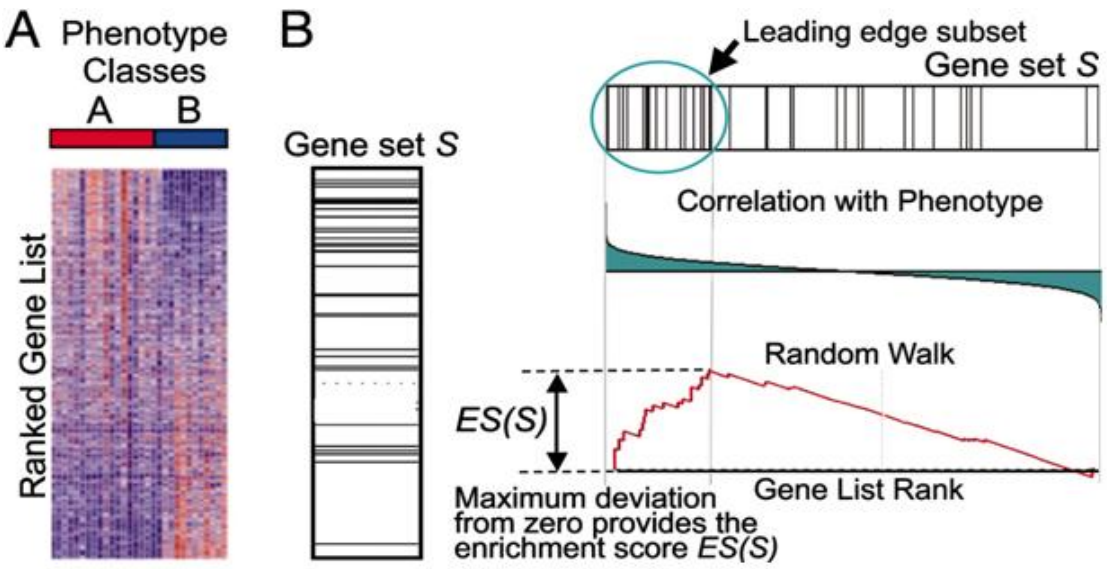


Figure 2.1: Overview of GSEA - A: example of gene expression profiles divided into classes; B: left: gene set S within the whole genome; up: leading edge subset of S; middle: correlation with the phenotype which can be computed by evaluating a t-statistic; down: plot of the running sum to compute the ES (picture taken from [5])

2.2 Computation of an Enrichment Score for the Gene Set S

Next we calculate an enrichment score (ES) that reflects the degree to which a gene set S is overrepresented in the extremes (top or bottom) of the ranked list L, i.e. if the set S contains a lot more upregeulated than downregulated genes, the enrichment scores of S will be high. The ES is calculated by walking down the list L, increasing a running sum when we encounter a gene in S and decreasing it when encounter a gene not in S. The magnitude of the increment depends on the correlation t_i of the gene with the phenotype (see also Figure 2.1). Finally the ES is the maximum deviation of the running sum to zero. Mathematically the running sum can be expressed by the difference of a weighted fraction of genes in S (P_{hit}) and the fraction of genes not in S (P_{miss}) with

$$\begin{aligned} P_{hit}(S,i) &= \sum_{\substack{g_j \in S \\ j \leq i}} \frac{|t_j|^p}{N_R}, \quad \text{where } N_R = \sum_{g_j \in S} |t_j|^p, \\ P_{miss}(S,i) &= \sum_{\substack{g_j \notin S \\ j \leq i}} \frac{1}{N - N_H}, \quad \text{where } N_H = |S|. \end{aligned} \tag{2.1}$$

Here p is a weighting exponent which is mostly set to zero (Kolmogorov-Smirnov-Statistic) or one. If $p \neq 0$ the null distribution of the ES (see point 3) will not be symmetric anymore. In this case it is conveniant to distinguish between positive and negative $ES(S)$.

2.3 Estimating Significance

In order to check whether there is a significant difference in the expressions profiles of the two phenotypes we have to check if $ES(S)$ is significant to a certain null distribution of enrichment

scores ES_{NULL}. To get this distribution we assume that the samples are randomly distributed among the phenotypes, which is our null hypothesis. We randomly permute the phenotype labels and recompute the ES (steps 1 and 2). Repeating this procedure many times leads to ES_{NULL}. A typical example of such a null distribution can be seen in Figure 2.2. Then we have to compare this with our observed $ES(S)$. We calculate a p-value which is the probability that an ES is more extreme than the observed $ES(S)$. If this p-value is low, the probability to observe an ES of at least or at most $ES(S)$ (corresponding to the sign of $ES(S)$) is small. If the p-value is below a certain threshold p^* we reject the null hypothesis, the observation is called significant and consequnetly the phenotypes show significant differences in their expression profiles.

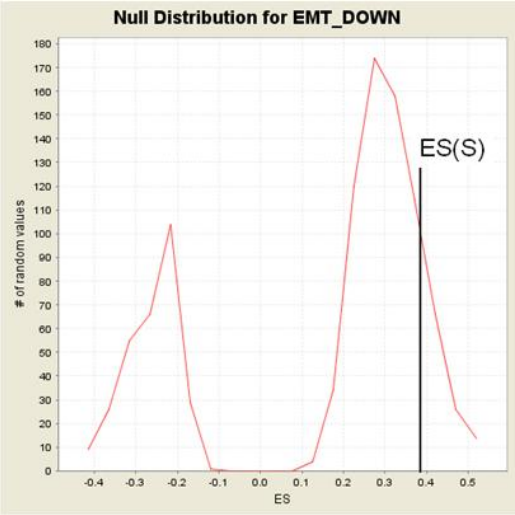


Figure 2.2: Null distribution of enrichment scores and high observed ES(S) which leads to significance (picture originally taken from [7])

2.4 Multiple Hypotheses Testing

If several gene sets $S_k, \quad k = 1, \cdots, m$ are being considered, we continue with the following procedure to control a so called false discovery rate q (FDR) which indicates the ratio of falsely rejected null hypotheses. First of all we determine $ES(S_k)$ for every involved gene set S_k and calculate its null distribution ES^k_{NULL}, the null hypothesis H_k and p-values p_k by performing the previous steps. We perform the following procedure, if applicable seperately for positive and negative $ES(S_k)$. Let $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$ be the ordered p-value, and denote by $H_{(i)}$ the null hypothesis corresponding to $p_{(i)}$. Then the FDR may be controlled at a pre-defined level q^* by the following procedure [1].

let k be the largest i for which $p_{(i)} \leq \frac{i}{m} q^*$;

then reject all $H_{(i)}, \quad i = 1, 2 \cdots, k.$

In the following section we describe how to compare lists of genes with known biological function (coming from databases) with lists of genes from QOE in such that we get more useful biological information.
The lists from Qlucore are generated by principal component analysis (PCA), so the lists are ranked. Then it would be great to use in the analysis the information coming from the fact that there is an order in the lists. One way to do this is explained in "Similarities for ordered gene lists" [6], where they are able to use this information about the order to compare lists.

Chapter 3

Our Approach

To compare the lists from the medical doctors and from Qlucore, we list and count their common elements. We compute the likelyhood that these lists share these elements assuming that Qlucore's list is the result of a random draw. If that probability is very small (below a given significance level), that means that these common elements "mean something". Qlucore's list matches with a doctors' list if the overlaps are statistically significant.

For a certain list given by Qlucore, we can search through all the doctor's lists and find the ones that match best.

3.1 Equal Probability

Firstwe assume that every gene has an equal probability to occur in a list and they are all independent from each other. Then the probability that a list from Qlucore, L_{QC}, and the list from medical doctors, L_{MD} have x elements in common if the lists are drawn randomly is given as

$$P(L_{QC} \cap L_{MD} = x) = \frac{\binom{m_1}{x}\binom{N-m_1}{m_2-x}}{\binom{N}{m_2}}, \tag{3.1}$$

where $\binom{n}{k}$ is the binomial coefficient. m_1 is the number of elements in the L_{MD} list and m_2 is the number of elements in the L_{QC} list. N is the total number of genes (22282). Note that when the length of the two lists are equal the probability in equation (3.1) is the same as the hypergeometric distribution.

$P(L_{QC} \cap L_{MD} = x)$ in equation (3.1) is computationally hard to evaluate for large values of m_1, m_2, x or N. The formula can be rewritten using that $a! = \Gamma(a+1)$ and that $log(\Gamma(a))$ can be computed as

$$\ln(\Gamma(a)) = -\ln(a) - \gamma \cdot a + \sum_{n=1}^{\infty}\left(\ln(1+\frac{a}{n}) - \frac{a}{n}\right), \tag{3.2}$$

where $\gamma \approx 0.57721$ is the Euler-Mascheroni constant [3, p. 157]. Then $P(L_{QC} \cap L_{MD} = x)$ can be rewritten as

$$\begin{aligned} P(L_{QC} \cap L_{MD} = x) = \exp(&\ln(\Gamma(m_1+1)) - \ln(\Gamma(x+1)) - \ln(\Gamma(m_1-x+1)) \\ &+ \ln(\Gamma(N-m_1+1)) - \ln(\Gamma(m_2-x+1)) \\ &- \ln(\Gamma(N-m_1-m_2+x+1)) - \ln(\Gamma(N+1)) + \ln(\Gamma(m_2+1)) \\ &+ \ln(\Gamma(N-m_2+1))) \end{aligned}$$

Thus it is practically possible to calculate the probability for large values of m_1, m_2, x and N.

3.1.1 Toy Example

We implemented this method in R. To check the code and the algorithm, two lists of length 1000 are randomly drawn from 22282 genes. In our example the two lists have 3 common elements which gives $P(L_{QC} \cap L_{MD} = 3) = 0.17$. A quite high probability, as expected since the lists are randomly drawn.

3.2 Different Probabilities

3.2.1 Model

We previously saw how to deal with the case where we assumed that all N genes appeared independently and with equal probabilities. These hypotheses are convenient and rapidly give results; nevertheless this is also clearly a very rough and unrealistic model. We will now consider the following model: the occurrences of different genes are independent but can now have different probabilities. Such a situation can arise for example if we consider a population for which certain genes are very common and other ones are rare (e.g. the genes coding for the synthesis of lactase are very likely to be found in European populations, not in Asian ones). For a certain gene G_i we will give a certain weight ω_i to its probability of being picked from a list of genes L. The total number of genes is quite large ($N \simeq 20000$). We will arrange them in different categories according to their respective probabilities. Let C be the number of such categories. Let ω_i be their respective probability weights and N_i the number of genes falling in each of these.

- $C = 1$ corresponds to the case we already dealt with (equiprobability).
- $C = N$ corresponds to the most general case where we give every gene its own probability.
- We will illustrate our approach with $C = 3$. We can think of these gene categories as "low probability", "medium probability" and "high probability" with weights $\omega_{low}(< 1)$, ω_{med} and $\omega_{high}(> 1)$. They respectively contain N_{low},N_{med} and N_{high} genes.

3.2.2 Defining Classes on Gene Lists

Once we have sorted the N genes in C categories, we classify the possible lists (i.e. sets) of genes of length m. We define classes (on the set of all possible sets of genes of size m) based on the number of elements in each of the C gene categories. All lists in a same class will be equivalent as regards likelihood computations. For example, let us consider lists of length $m = 2$ with $C = 3$ categories. The aforementioned classes would be the $\{\overline{L_{(x_{C_1},x_{C_2},x_{C_3})}}\}$, the classes whose elements have respectively x_{C_1},x_{C_2} and x_{C_3} elements in ("high probability category","medium probability category","low probability category") such that $x_{C_1}+x_{C_2}+x_{C_3} = 2$. We can sort all the $C_N^2 = \binom{N}{2}$ lists (of form $\{G_i, G_j\}$) into one of these few classes: $\{\overline{L_{(2,0,0)}}, \overline{L_{(1,1,0)}}, \overline{L_{(1,0,1)}}, \overline{L_{(0,1,1)}}, \overline{L_{(0,2,0)}}, \overline{L_{(0,0,2)}}\}$. How many such classes are there ? For a list L of length m and a given number of categories C, it is equivalent to counting the number of tuples $(x_{C_1}, \ldots, x_{C_C}) \in \mathbb{N}^C$ such that $x_{C_1} + \ldots + x_{C_C} = m$. We can show that the number of such tuples, and hence the number of classes is $C_{m+C-1}^m = \binom{m+C-1}{m}$, which is $O(m^{C-1})$. Typically $m \simeq 100$. We see that considering the most general case of N genes with N different probabilities is computationally not doable. Considering all classes is numerically feasible for relatively low values of C. The advantage of working with these classes is that we can simplify the general analysis and reduce our computations to a reasonable number of cases.

3.2.3 Probability of a Class

Let $\overline{L_{(x_{C_1},x_{C_2},x_{C_3})}}$ denote the class of gene lists such that x_{C_1} genes are in the "high probability" category, x_{C_2} in the "medium probability" and x_{C_3} in the "low probability". As all the lists in a class are equivalent, they have the same probability and we can hence define the probability of a

class. The following formula gives the probability of finding a list of a certain class when we pick a set of m genes.

This is a Multivariate Fisher's Noncentral Hypergeometric Distribution: Whenever the next formula is defined, we have

$$\mathbb{P}(L_{(x_{C_1},x_{C_2},x_{C_3})}) = \mathbb{P}(\overline{L_{(x_{C_1},x_{C_2},x_{C_3})}}) = \frac{C_{N_{High}}^{x_{C_1}} \cdot C_{N_{Med}}^{x_{C_2}} \cdot C_{N_{Low}}^{x_{C_3}} \cdot \omega_{High}^{x_{C_1}} \cdot \omega_{Med}^{x_{C_2}} \cdot \omega_{Low}^{x_{C_3}}}{\sum\limits_{\substack{(i,j,k)\\ i+j+k=m}} \left(C_{N_{High}}^{i} \cdot C_{N_{Med}}^{j} \cdot C_{N_{Low}}^{k} \cdot \omega_{High}^{i} \cdot \omega_{Med}^{j} \cdot \omega_{Low}^{k} \right)}$$

When this formula is not defined, we have $\mathbb{P}(L_{(x_{C_1},x_{C_2},x_{C_3})}) = \mathbb{P}(\overline{L_{(x_{C_1},x_{C_2},x_{C_3})}}) = 0$

We can compute all the $C_{m+C-1}^{m} = O(m^{C-1}) = O(m^2)$ values quite quickly. We evaluate all the possible numerators with a complexity $O(m^2)$, then we sum them to get the normalizing denominator and get all classes' probabilities.

Furthermore these computations can be very easily parallelized to save time. Use of GPGPU (CUDA...) for the implementation would certainly be of great interest.

3.2.4 Intersection of a Random List with a Given List

Now consider that we have a fixed list of genes L_1 of length m_1. We now pick a second list L_2 of length m_2. Let us compute the likelihood of finding x common elements in those two lists.

Knowing L_1, we know $(N_{L_1}^{low}, N_{L_1}^{med}, N_{L_1}^{high}, N_{L_1^c}^{low}, N_{L_1^c}^{med}, N_{L_1^c}^{high})$ the number of genes from L_1 or from L_1^c that fall in each of the 3 probability categories
"High/Medium/Low probability".

We can then define $\overline{L_{(x_{L_1}^{low},x_{L_1}^{med},x_{L_1}^{high},x_{L_1^c}^{low},x_{L_1^c}^{med},x_{L_1^c}^{high})}}$ the class of sets of m_2 genes such that $(x_{L_1}^{low}, x_{L_1}^{med}, x_{L_1}^{high}, x_{L_1^c}^{low}, x_{L_1^c}^{med}, x_{L_1^c}^{high})$ genes fall in each of the 6 aforementioned categories:
"High/Medium/Low probability" and "In/Out of L_1".

As in (3.2.3), we have a Multivariate Fisher's Noncentral Hypergeometric Distribution: whenever this formula is defined we have

$$\mathbb{P}(\overline{L_{(x_{L_1}^{low},x_{L_1}^{med},x_{L_1}^{high},x_{L_1^c}^{low},x_{L_1^c}^{med},x_{L_1^c}^{high})}}) = \frac{1}{\sum\limits_{\substack{possible\\ \overline{L_{(x_{L_1}^{low},\dots,x_{L_1^c}^{high})}}}} \prod C_{N_*^{\cdots}}^{x_*^{\cdots}} \omega_{\cdots}^{x_*^{\cdots}}} \cdot \prod_{i=1}^{6} C_{N_*^{\cdots}}^{x_*^{\cdots}} \omega_{\cdots}^{x_*^{\cdots}}$$

When this formula is not defined, we have $\mathbb{P}(\overline{L}) = 0$

Complexity is now (for the same reasons as before) $O(m_2^{2 \cdot C-1})$ which is still reasonable for "small" values of C. For $C = 3$, $O(m_2^5)$. We can again use a parallelized implementation to speed up the computations.

We will denote by X the number of common elements between our given list L_1 and our "new" random list L_2. We are interested in computing the values of $\mathbb{P}(X = x)$ for $x = 1, \dots, \min(m_1, m_2)$.

We have $\{X = x\} = \{x_{L_1}^{low} + x_{L_1}^{med} + x_{L_1}^{high} = x\}$. Hence for a given L_1,

$$\mathbb{P}_{L_1}(X = x) = \sum_{\{\overline{L_{\dots}} | x_{L_1}^{low} + x_{L_1}^{med} + x_{L_1}^{high} = x\}} \mathbb{P}(\overline{L_{(x_{L_i}^{low},x_{L_i}^{med},x_{L_i}^{high},x_{L_i^c}^{low},x_{L_i^c}^{med},x_{L_i^c}^{high})}}) \tag{3.3}$$

We have $Card(\{\overline{L_{\dots}} | x_{L_1}^{low} + x_{L_1}^{med} + x_{L_1}^{high} = x\}) \leq x^3 \leq m^3$. Hence this is a sum of a very reasonable number of terms and can be computed in a reasonable time.

3.2.5 Computation of $\mathbb{P}(X = x)$

As we explained before, all lists in a given class are equivalent as regards probability computation, so we can write that $\mathbb{P}_{L_i}(X = x) = \mathbb{P}_{\overline{L_i}}(X = x)$

Hence we finally get

$$\mathbb{P}(X = x) = \sum_{\substack{possible \\ \overline{L_i}}} \mathbb{P}(\overline{L_i})\mathbb{P}_{\overline{L_i}}(X = x) \tag{3.4}$$

As explained in (3.2.2) this last sum involves a reasonable number of terms. Assuming $m = m_1 \simeq m_2$, we have : $O(m^{C-1} = O(m^2)$ terms. Each term "costs" $O(m^{2C-1}) = 0(m^5)$. Total cost will be $0(m^{3C-1}) = O(m^7)$; The speed can be dramatically improved by adequately parallelizing all the different parts involved in the computation of the sum.

3.2.6 Summing up

We are now able to compute the likelihood that two lists of genes L_1 and L_2 have a certain number of elements in common in the case where genes have different probabilities of appearing.

We have an explicit formula enabling us to deal theoretically with the most general case (every gene has its own likelihood) and that can be used practically to a slightly simplified case (genes grouped in several likelihood categories).

Furthermore the formulas we need to evaluate require computations that can be parallelized easily. This should permit an efficient implementation using GPGPU.

3.3 Results

Using the methods proposed in section 3, we compared different files containing gene sets with known biological functions to lists of genes that were found to contain the biggest variations across children with leukemia. The lists of genes that varied the most were found by the software Qlucore, solely based on statistical methods without imposing any assumptions.

The dataset used contains information on gene expression for children with leukemia. This data was run through Qlucore - without any assumptions regarding distributions or anything - found the genes that varied most for these children. The list given by Qlucore was then compared to gene sets with known biological functions using the hypergeometric distribution (the first method we proposed). The probabilities of finding the number of common elements observed in the lists are given in table 3.1, from which we see that the probabilities of seeing the number of common elements actually observed on these lists would be very small if the lists were drawn randomly. This implies that the genes responsible for lymphocyte activation and differentiation are overrepresented among the genes that vary most for children with leukemia. This makes intuitive sense, as white blood cells' functioning is disrupted in patients with leukemia, lending credibility to our approach.

The results indicate that our method may indeed be used to single out genes responsible for disease, and hence find candidates at which future drugs could be targeted.

	T-ALL_downregulated	T-ALL_upregulated
Lymphocyte activation	7.822e-03	5.280e-13
Lymphocyte differentiation	5.313e-02	8.159e-05

Table 3.1: Probabilities of seeing the number of elements in both lists that was actually observed between the genesets given in the row names and lists produced by Qlucore

Recommendations

Based on the work presented in this report, we expect that the road embarked upon will result in a method giving good indications as to the genes involved in diseases.
Using the preliminary approach described in 3, we were able to identify genes that are likely to be involved in a certain disease (leukemia in the example). This should of course be tested in several experiments before we know for sure that our results are in accordance with reality.
If experiments confirm our results, and show that the genes we identified do play an important role in disease, then this method could be used to find candidates for gene targeting and hypothetical cures.
Extensions of the approaches presented in section 3 could involve dropping the assumption that genes are drawn independently. The definition of 'intersecting elements' on the two lists being compared could also be extended: we could look at the commonality of gene groups instead of looking at single genes. This way, pairs or larger sets of genes as a whole could be seen as typical of a certain 'population'. Hence it would be important to find the probability of a set of intercorrelated genes being on two lists instead of the probability of several independent genes appearing on the lists.

Conclusion

In this report, we point out that various databases of genes contain lots of information, which can be very beneficial when used in connection with other experiments. Also, we describe how to use the databases, i.e. how to access the information they contain.
We also suggest a way of using the information together with other data. We saw how to find the probability that a certain number of genes is shared by two lists (possibly of different lengths) assuming one of them is drawn randomly. We also extend this to a more realistic model without the assumption of equal probabilities for all genes and propose a way of limiting the computational cost in this case.
Furthermore, a simple program was implemented to show that the idea works and different sets of data were run through it. This resulted in the identification of a gene set with a known biological function that were also the genes that varied most for diseased patients. Candidates for further investigation into the disease could thus be identified.
Our approach seems very promising but further experimentation would be needed to confirm this impression.

Bibliography

[1] Y. Benjamini and Y. Hochberg: *Controlling the false discovery rate: A practical and powerful approach to multiple testing*, J. Roy. Stat. Soc. B Met., 57(1):289-300, 1995.

[2] B. Efron and R. Tibshirani: *On testing the significance of sets of genes*, Ann. Appl. Stat., 1(1):107-129, 2007.

[3] Krantz, S. G. : *The Gamma and Beta Functions*, in Handbook of Complex Variables, Birkhuser, pp. 155-158, 1999

[4] L. Sachs: *Angewandte Statistik: Anwendung statistischer Methoden*, Springer Verlag, seventh edition, 1992.

[5] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukheryee, B. L. Ebert, M. A. Gilette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov: *A knowledge-based approach for interpreting genome-wide gene expression profiles*, Proc. Natl. Acad. Sci. USA, 102(43):15545-15550, Oct. 2005.

[6] Yang, X. et al.: *Similarities of ordered gene lists*, J. Bioinf. Comp. Biol., 2006.

[7] *http://www.broadinstitute.com*

[8] *http://www.qlucore.com/home.aspx*