U N I V E R S I T Y O F T A R T U
Faculty of Mathematics and Computer Science
Institute of Computer Science

Darja Kruševskaja

Annotation Driven Hierarchical Clustering Analysis

Master Thesis

Supervisor: Jaak Vilo, PhD

TARTU 2007

Contents

Chapter 1

Introduction

1.1 Motivation and Background

Due to the evolution of hardware technology computer systems are able to store huge amounts of multidimensional data. Development of the technology has affected the course and tempo of molecular biology research; a variety of the experiments conducted in laboratories can produce large amounts of data. To process these huge quantities of data, analytical softwareis needed.

Back in 1986 an international collaborative research program the Human Genome Project, was set up. The goal of this project is the complete mapping, as well as understanding the functions of the genes of human beings [Offf]. The International Human Genome Sequencing Consortium published the first draft of the human genome in the *Nature* journal in February 2001 with the sequence of the entire genome's three billion base pairs some 90 percent complete [LLB+01]. At the moment genomes of yeast, fruit fly, chicken, mouse, dog, chimpanzee, etc. are available [GBB+96, ACH+00, HMB+04, WLTB+02, KBH+03, CVS+05]. At present, the genoms of 51 eukariotic organisms have been sequenced [Offb].

The genome includes both the genes and the non-coding sequences of DNA. Once a genome has been sequenced, determining a gene, is one of the first and the most important steps in understanding the genome of a species once it has been sequenced. However, determining that *a sequence is functional* should be distinguished from determining *the function of the gene or its product*. The function of the gene has to be studied separately:

it demands *in vivo* experimentation (lab tests) through gene knockout and other assays. Nowadays, gene function can also be predicted using bioinformatic methods. In the case of some genes, function prediction remains a very difficult task, independent if the methods used.

Earlier, each gene was studied separately. Special *in vivo* experiments were performed to determine the possible role of the gene. Once the function of the gene was determined and confirmed by experiments, the corresponding article was published. This approach to the decoding process was ineffective: it was slow, laborious and expensive; the information was distributed, unstructured and the common terminology was missing.

The evolution of experiments that can be performed to study biological data has changed the course of research. High throughput technologies, such as microarray analysis, made it possible to track thousands of observations of different gene tests at the same time. Today, microarray experiments are widely used in biology to get information about the cellular states. Each experiment may contain 40,000+ measurements characterising the behaviour of different genes under different conditions. The data derived from microarray experiments is referred to as *gene expression data*.

The lack of common terminology and distribution of knowledge slowed down the speed of researches, and made it difficult to search for information already available. Projects like the Gene Ontology (GO) sprang up to organize available information in order and work out structured and controlled vocabularies [Offc, ABB+00]. The vocabularies are used to annotate the biological knowledge or a predicted characteristic for a given gene. GO is a structured network that consists of defined terms and the relationships between them. The relations describe three attributes of gene products: *Molecular Function, Biological Process* and *Cellular Component* [Lew05]. Nowadays, GO is a very popular resource and many research communities use GO terms for attaching biological information to genomic elements [CMB+04].

1.2 State of the Art

The amount of data produced by new technologies has also affected the methods that can be used to learn new things about the genome. A key initial step in the analysis of gene expression data is *clustering*. A clustering problem consists of elements and a characteristic vector for each element. A

measure of similarity is defined between pairs of such vectors. The goal of clustering is to partition the elements into subsets, which are called *clusters*, in such a way that elements are similar withing the cluster and dissimilar between clusters [SES01]. While clustering gene expression data, the genes that exhibit similar behaviour are put into the same cluster. Because of the large number of genes, clustering is useful exploratory technique of gene expression data [YHR01].

In spite of the fact that hierarchical clustering is computationally expensive, it is often used for clustering the expression data [BV00, ESBB98]. The advantage of hierarchical clustering analysis is that it provides the **structure** for the whole data set. Hierarchical clustering builds a hierarchy of clusters which can be represented as a binary tree. The individual elements of the clustered data set form the leaves, while the root of the tree represents the whole data set.

After clustering, the analysis is usually continued. The user finds the cluster of interest and studies it independently of the data set. The choice of this cluster may be based on different criteria: the cluster contains some particular genes, or the cluster of genes that behave very similarly. Annotated genes can be clustered together with non-annotated genes. This approach can further understanding of the functions of many genes for which information has not been previously available. Furthermore, genes with similar behaviour in the same cluster are likely to be involved in one and the same process, and a strong correlation between those genes indicates co-regulation [JTZ04].

But even the annotation of all genes in the organism does not explain the functioning of the genome as a whole. A process in the organism that might seem to a person as a single unit, could possibly involve a large number of genes, and hence be considered as a chain of reactions. Some projects consider biological information from exactly this point of view. For instance, KEGG PATHWAY database organises data using the interactions between molecules and reaction networks [Offe, KGH+06].

While performing an analysis, the main goal of a researcher is to acquire the deep understanding of **the structure** in the data set combined with **biological understanding** (similar expression profiles, shared functions) that would help in selecting the further analysis.

As we have mentioned before, the number of clusters depends on the number of items in the data set. For instance, the number of genes in a human genome is around 30,000. Thus, the hierarchical tree would

contain 29,999 internal nodes and 30,000 leaves. Presenting such data in textual or tabular formats is not suitable for human interpretation. Instead, visualisation techniques should be used. However even a picture that contains a huge tree is hard for a human to comprehend. First of all, it is difficult to grasp such a picture at once. Secondly, it is even more difficult to find a cluster of interest.

One of the possible solution to the problem are *collapsed nodes*, that would each give a summarised view of the corresponding subtree. Usually, collapsing is performed on subtrees that are either at a fixed depth or o a fixed size. The disadvantage of diminution is that the collapsing process does not take into account the features of the underlying data.

The goal of this theses is to find a representation that is compact enough, but, at the same time, preserves the structure of the data set and captures the necessary level of details.

1.3 Contribution and Overview

We propose a new annotation driven hierarchical clustering analysis that tries to find the right balance between the raw data, the clustering results, and the biologically most interesting features in the data. The technique takes an advantage of the annotation data that is publicly available.

Our technique works on data sets where each object is described by two sets of properties. These sets are used separately, at different stages of analysis: the gene expression data is used for hierarchical clustering, whereas the annotation set is used for collapsing.

The whole technique consists of four steps. The first step includes clustering the genes by their expression profiles using a fast approximate hierarchical clustering algorithm [Kul04]. Secondly, the annotation terms for all the subtrees in hierarchical clustering are found using g:Profiler [RKP+07]. Thirdly, an automated analysis of clusters is performed. The analysis searches for the subtrees with significant over-representation of some feature, and passes them to the collapsing process. Finally, the results are presented as a compact navigable graphical tree.

The tool, that was developed as a part of the master theses, implements this technique and demonstrates its capabilities by the example of gene expression data.

In what follows, we give an overview of the thesis. Chapter 2 provides

the introduction to the biological problem and gives an overview of the background. Chapter 3 reviews the existing visualisation approaches, their advantages and disadvantages. At the end of the chapter, we also give a short overview of our solution and briefly explain why it is potentially better then existing ones. Chapter 4 introduces the technique and explains the analysis step by step.

Chapter 5 introduces the tools and algorithms that are integrated by the implementation of the technique. The chapter also lists the data set types that are used as the input of the implementation. The practical result of this thesis is Treeviewer, a tool for hierarchical clustering analysis and visualisation. It is described in chapter 6. The latest version of Treeviewer can be found at `http://emu.at.mt.ut.ee/treeviewer/`. Final chapter, Chapter 7, contains the conclusion and the summary in Estonian.

Chapter 2

Fundamentals

In this chapter we provide an introduction to the biological problem. The Gene Ontology terms, biological pathways and data annotations are discussed. The biological background is followed by the introduction to clustering.

Biologists often want to analyse gene expression data to understand and discover the various relationships between items. According to Gilbert *et al.* the usual approach to reach the goal is to [GSvH00]:

1. cluster the items by their expression profiles;
2. display the result in some visually meaningful way.

2.1 Microarray technology

Proteins are the components of cells and tissues. They participate in many processes within a cell: biochemical reactions, structural or mechanical functions, signalling, immune responses, cell adhesion or cell cycle. They are assembled from amino acids using information encoded in genes. Protein production from genes involves 2 stages: transcription and translation. During the transcription, a single strand of the messenger ribonucleic acid (mRNA) is copied from DNA segment, coding the gene. After that the translation starts. During it mRNA is used as a template to assemble a chain of amino acids to form a protein [Par03].

Gene expression investigates the amount of transcribed mRNA in biological system [Par03]. When gene is expressed the corresponding protein is processed. The gene expression level depends on the tissue type and several conditions, like stress level, nutrients etc. The expression levels for a gene under different conditions form *gene expression profile*.

The microarray technology enables to measure expression levels of genes [TSS06]. Natural Human Genome Research Institute defines microarray technology as "a new way of studying how large numbers of genes interact with each other and how a cell's regulatory networks control vast batteries of genes simultaneously." [Offf]

The principle of microarray work is based on the distinctive feature of the DNA: DNA is doublestranded and consists of two complementary sequences. Under the normal conditions the complementary and the single-stranded sequences will combine into the single molecule. This process is referred to as *DNA hybridization*. Microarray quantifies gene expression by measuring the level of hybridisation of single-stranded DNA sequences, fixed on a small glass of nylon matrix, and mRNA representation from the sample under study [Par03]. A single experiment is performed in each position (spot) of the matrix.

There are two technologies for DNA microarrays. In the first, short sequences of nucleotides are synthesised onto a slide or attached after synthesis. In the other, complementary DNAs (cDNAs) are deposited onto a slide, called spotted DNA microarrays [Par03].

The data sets used in current theses are produced by spotted DNA arrays. In these mRNA sequences from two different biological samples are reverse-transcribed into cDNA and labelled with two different fluorophores, usually red and green. Then, these samples are mixed and hybridized to the microarray that is then scanned [TSS06]. Figure 2.1 illustrates the microarray experiment.

A laser scanner measures dye of each fluorophores. Brighter colour indicates higher amounts of hybridized cDNA, which in turn indicates higher gene expression. Measurement of relative gene expression across two samples is possible due to the use of two channels.

The result of the microarray experiment is usually visualised as it is shown in Figure 2.2. Every spot represents a single experiment. The colour of the spot illustrates the expression level. If neither of both kinds of sequences have reached the spot than it is of a black colour, if both have reached it, than the colour of the spot is yellow. If only one kind

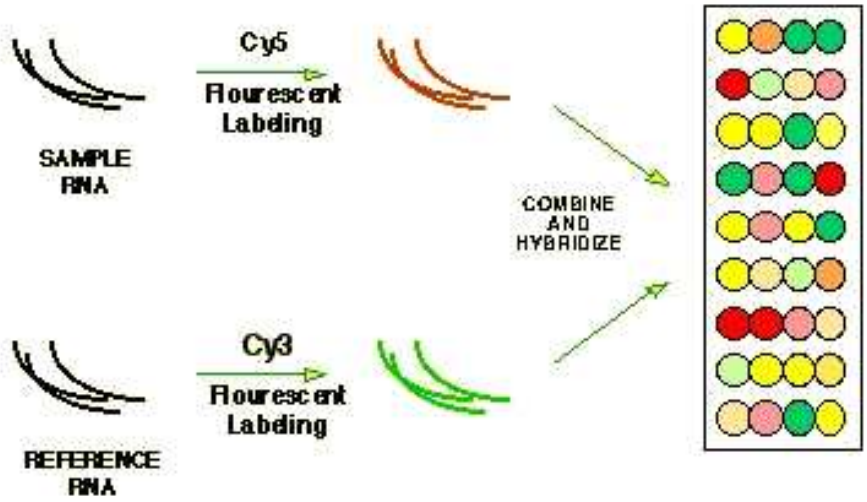


Figure 2.1: Overview of labelling and hybridisation in microarrays.

of sequences have reached the spot, then it is correspondingly of red or green colour. The proportion or the number of sequences that reached the spot can be different, this influences the shade and depth of the colour. [DTW05].

The high volume of such data has emphasised the need for statistical and data analytic techniques. Two computational issues are associated with microarray analysis: procession of experimental data and interpretation of it. The result of the data procession is a *gene expression matrix E*. For instance, it may be composed of k rows, each corresponding to a gene g_i on a microarray, and m columns each corresponding to a condition τ_j for which expression levels were measured. The element e_{g_i,τ_j} represents the expression level of gene g_i under condition τ_j [DTW05]:

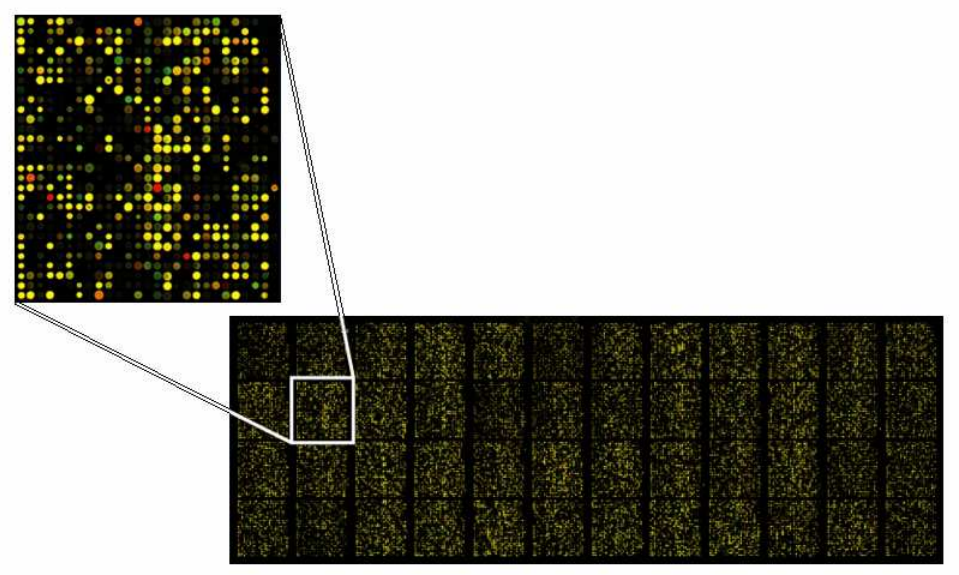


Figure 2.2: Example of 40,000 probe spotted onto microarray with enlarged inset to show details.

$$E = \begin{pmatrix} e_{g_1,\tau_1} & e_{g_1,\tau_2} & \dots & e_{g_1,\tau_m} \\ e_{g_2,\tau_1} & e_{g_2,\tau_2} & \dots & e_{g_2,\tau_m} \\ \vdots & \vdots & \ddots & \vdots \\ e_{g_k,\tau_1} & e_{g_k,\tau_2} & \dots & e_{g_k,\tau_m} \end{pmatrix}.$$

2.2 Clustering

Clustering is the unsupervised classification of objects into groups or *clusters*. The goal of clustering is to put similar objects to the same cluster and dissimilar ones to different clusters. Usually objects are described and clustered using a set of features or attributes [MS99]. Based upon the collection of attributes of these objects, similarity measure evaluates how similar objects are [DTW05].

A very important step in any clustering is a choice of a distance measuring method. The similarity between two objects will be measured using it. There are very many possibilities to calculate distance between two objects: one can use widely known distance measures, such as Euclidean

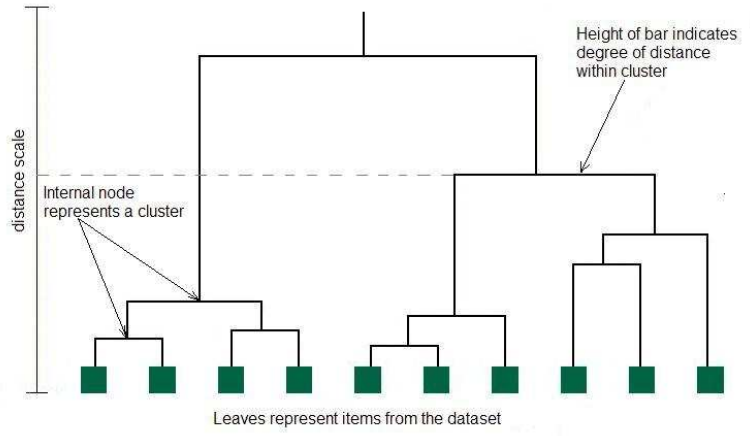


Figure 2.3: The example of a dendrogram. Leaves of the tree represent items from the data set, internal nodes represent a cluster. The height of bar indicates degree of distance within a cluster.

distance [BV00]. One can also propose some new measurement, that will be the most effective in that situation. The choice of distance measuring technique is highly dependant on the goal, the application and the data itself.

There are many different clustering algorithms, but they can be grouped into two basic classes: *hierarchical clustering* and flat or *non-hierarchical clustering*. As a result of non-hierarchical clustering objects are divided into different groups, and the relations between groups are undetermined. These algorithms often have iterative nature, they start with some initial state and at each step of iteration they improve clusters until converged [MS99].

Hierarchical clustering, unlike non-hierarchical one, provides data structure that represent the relation between all objects in the data set. Hierarchical clustering consists of successive joining together objects or group of objects based upon the measure of similarity or distance between the objects [DTW05]. In other words, it works by iteratively joining the two closest clusters starting from singleton clusters. The result of the process is a tree or a *dendrogram* where each leaf represents an object from the data set. Each internal node represents the cluster that contains all objects of its descendants [MS99]. The example of the dendrogram can be seen in Figure 2.3

The implementation of clustering algorithm can be characterised also from other aspects. The most relevant to this work are:

- Hard vs. fuzzy: In a hard clustering item can be assigned to exactly one cluster. Fuzzy clustering allows the membership in multiple clusters by introducing a membership function $W_{i,j}$ between each cluster-item pair to measure the degree of association.

- Agglomerative vs. divisive: This aspect relates to the algorithmic structure and operation order. Agglomerative approach begins with clusters, each containing only one object, and successively mergers clusters together, until a stopping condition is satisfied. Divisive approach is the opposite to agglomerative. It starts with the whole data set and iteratively partitions clusters [JMF99].

Another important property in clustering is distance between clusters. *Linkage* is the criterion by which the clustering algorithm determines the actual distance between two clusters. There are three types of linkage, and thus three ways to calculate the distance:

1. *Single Linkage*: The distance between two clusters is the minimum distance between members of the two clusters;

2. *Complete Linkage*: The distance between 2 clusters is equal to the greatest distance between a member of cluster i and a member of cluster j;

3. *Average Linkage*: The distance between clusters is calculated using average values of a pair of clusters.

Gene expression matrix can be considered as a data set where each data item has got a set of properties. In other words each gene g_i can be expressed as a vector $g_i = (e_{g_i,\tau_1}, e_{g_i,\tau_2}, ..., e_{g_i,\tau_m})$. Thus one can calculate the distance between expression profiles of a genes pair. This makes it possible to apply clustering algorithms to the gene expression data.

Due to the nature and size of the microarray results, clustering is often used to perform exploratory analysis. Clustering allows to group genes with similar expression profiles together and present the summarised result to the researcher. The main goal of clustering is to find and represent a clear and understandable structure of the data set.

x_i	attr1	attr2	attr3
x_1	9	3	7
x_2	10	2	9
x_3	1	9	4
x_4	6	5	5
x_5	1	10	3

Table 2.1: The sample data set consists of 5 items. Each item is described by three attributes.

Despite the fact that hierarchical clustering algorithms are computationally expensive, they are usually preferred over the flat algorithms. The bottleneck of efficiency problem lays in the calculation of distances between all the pairs in the data set. However, some algorithms have already overcome this problem [Kul04]. The advantage of the hierarchical algorithms is that they provide more information about the data set as a whole.

Let us consider an example of an agglomerative average linkage clustering algorithm. The sample data set consists of 5 items, each item can be described using the set of attributes: attr1, attr2 and attr3. Table 2.2 contains data items from the sample data set with attribute values.

During the first step of the hierarchical clustering, each object is placed in its own cluster. As a result we obtain 5 singleton clusters. Then a list of pairwise distances is constructed. Euclidean distance is used as a distance measure.

Euclidean distance d between two items x_1 and x_2 can be calculated using the following formula:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{i,k} - x_{j,k})^2},$$

where m is the number of attributes and $x_{h,k}$ is the value of attribute k of the item x_h. For instance, the distance d between x_1 and x_2 is calculated as following:

x_i \ x_j	x_1	x_2	x_3	x_4	x_5
x_1	0	2.5	10.44	4.12	11.75
x_2	2.5	0	12.5	6.4	13.93
x_3	10.44	12.5	0	6.48	1.41
x_4	4.12	6.4	6.48	0	7.35
x_5	11.75	13.93	1.41	7.35	0

Table 2.2: Pairwise distances for the sample data set. The distances are calculated using Euclidean distance measure.

$$d(x_1, x_2) = \sqrt{(9-10)^2 + (3-2)^2 + (7-9)^2} \approx 2.5;$$

The pairwise distances for the sample data set are listed in the Table 2.2.

Ar the next step the most similar pair of clusters has to be found. In this case it is (x_3, x_5), because the distance between these two is the smallest:

$$d(x_3, x_5) \approx 1.41.$$

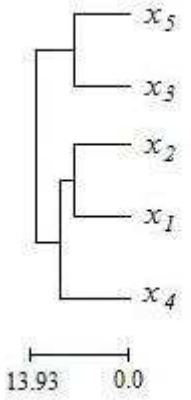
The clusters x_3 and x_5 are joined together, and from now on have to be considered as a single cluster. The distances between freshly built cluster and other clusters are recalculated using average linkage algorithm.

We continue to join the closest clusters and recalculate distances between clusters until all clusters are connected and the only one cluster is left (Figure 2.4).

2.2.1 Hierarchical Clustering Data Visualisation

As we have already discussed, the expression data is usually clustered hierarchically. Hierarchical clustering algorithms output binary trees, in which each leaf presents the data item from the initial data set and the internal node represents a hierarchical cluster. The interval of possible values for the distance depends on the distance measure.

Although clustering methods can usually organise the tables of gene expression measurements, the resulting ordered output still can be a massive collection of numbers. Therefore, clustering methods are usually combined with a graphical representation of clustering result. The end result



x_i	attr1	attr2	attr3
x_5	1	10	3
x_3	1	9	4
x_2	10	2	9
x_1	9	3	7
x_4	6	5	5

Figure 2.4: Dendrogram illustrates the result of the agglomerative average linkage clustering.

is the representation of the complex gene expression data that, through statistical organisation and graphical display, allows researches to explore the data in an intuitive manner [ESBB98].

The result of the hierarchical clustering can be naturally represented by a tree. But there are two main problems associated with it [GSvH00]:

- displayed structure of the tree must be uniquely determined;
- trees do not scale up for large data sets.

Thus, effective clustering visualisation technique must solve both of these problems.

2.3 Biological Data Annotation

2.3.1 The Gene Ontology Project

The level of genetic knowledge has dramatically increased during last few years. From this evolved a need for working out a common dictionary of terms and organising these terms in a structured way.

The Gene Ontology (GO) project[1] provides a controlled domain that describes genes and gene product attributes in any organism. The GO

[1] `http://www.geneontology.org/`

Project develops three structured and controlled vocabularies or *ontologies*. They cover the following areas [ABB+00]:

- **Cellular component** or component of a cell, e.g. nucleus.
- **Biological process**, that means one or more sequential assemblies of molecular function. E.g. signal transduction.
- **Molecular function** or activities that occur at molecular level. E.g. catalytic activity, transporter activity.

To develop the ontologies the GO project works on three different aspects. The first aspect is the maintenance and development of vocabularies. The second aspect covers the *annotation* of gene products, the characterisation of gene products using terms from the ontologies. Third, the GO project develops several tools that allow to use the ontologies and annotations [ABB+00].

The GO vocabulary consists of terms. Each GO term has a unique alphanumerical identifier (*e.g.* GO:123456), a common name, synonyms (if applicable), and a definition. Each term belongs to only one of three ontologies.

The ontologies are structured as directed acyclic graphs, where each child term can have many parents. The child term is more specialised then its parent. The terms in an ontology are linked by two kinds of relationships [Offc]:

- **is_a** is a class-subclass relationship;
- **part_of** means that one class is a part of another class.

New terms and annotations are suggested by members of research and annotation communities. Once submitted, they are reviewed by members of the GO consortium to determine their applicability.

The example of GO annotation tree demonstrating relations between terms can be found in Figure 2.5. This part of vocabulary corresponds to the biological process of cholesterol and carotenoid byosyntetic process.

2.3.2 Biological Pathways

According to Karp biological pathway is a linked set of biochemical reactions, linked in the sense that the product of one reaction is a reactant of,

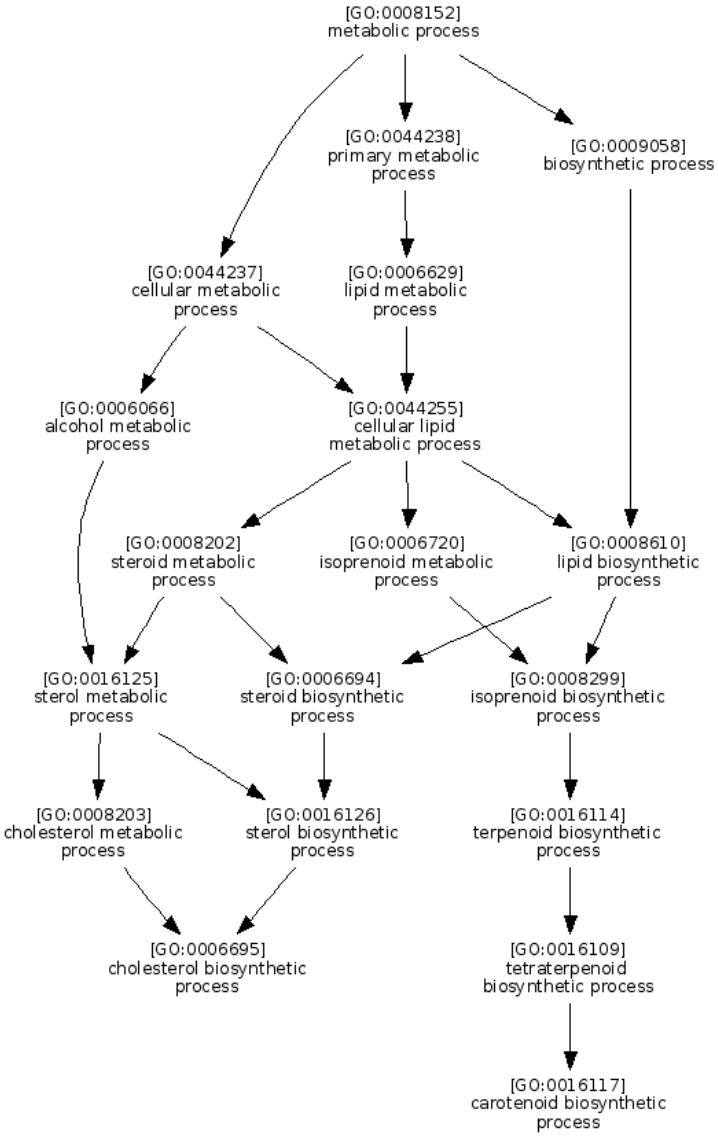


Figure 2.5: A fragment of GO hierarchical vocabulary. It corresponds to the biological process of cholesterol and carotenoid byosyntetic processThis picture is derived from g:Profiler.

or an enzyme that catalyzes a subsequent reaction [Kar01]. In other words biological pathway represents a network of reactions. Each network shows how the biological function is accomplished by describing the interaction of molecules. Pathways enable to store the information about biological processes (See Figure 2.6 for example.). They also allow to build hypotheses, integrate knowledge from literature, capture empirical results, share information and simulate processes [SND05].

The sequence of steps comprising a pathway is rarely linear. Single step of the reaction often requires multiple input, it also may produce multiple outputs. The pathway may contain redundancy: multiple parallel chains can lead to the same result. Conversely, an individual step can be multifunctional and be involved in several pathways. Pathways can block or vice versa run each other[Sch04].

A natural representation of a pathway is a directed graph. Edges in the pathway represent cause/effect dependencies that are hold among molecules [Sch04].

A pathway database is a bioinformatic database that describes biochemical pathways and their component reactions, enzymes, and substrates. Historically, they arose at the intersection of genomics, biochemistry, database systems, and artificial intelligence. Biochemists have had a long-standing effort to catalogue the catalytic activities of known enzymes in printed form [Kar01]. But, the amount of knowledge has grown, thus nowadays relational databases are the most common to store pathway data. Most pathway databases created to date describe metabolic pathways. There are several bigger databases that contain pathway information about different species. In the current work Kyoto Encyclopedia of Genes and Genomes[2] (KEGG)[KGH+06, Offe] is used as one source of annotating genes.

KEGG is a "biological systems" database that integrates both molecular building block information and higher-level systematic information. Molecular building blocks are distinguished between genetic building blocks (KEGG GENES) and chemical building blocks (KEGG LIGAND), while the systemic information is represented as molecular wiring diagrams (KEGG PATHWAY), hierarchies and relationships among biological objects (KEGG BRITE) [KGH+06].

In the current work we are using data that is originally provided by

[2] http://www.genome.jp/kegg/

KEGG PATHWAY. At the moment it contains 48,581 pathways that are generated from 317 reference pathways. It holds manually drawn pathway maps that represent the knowledge on the molecular interaction and reaction networks for metabolism, other cellular processes, and human diseases [KGH+06].

KEGG PATHWAY has a structure of a nested graph. The nested graph is a graph where nodes can themselves be graphs. Thus it can be considered as a hierarchical structure. For instance class metabolism has several subclasses: *Carbohydrate Metabolism*, *Energy Metabolism*, etc [Offe]. Every pathway has it is own 5 digit identifier, name and the date of creation. For instance the pathway with code 00051 was created on February 5, 2007 and represents fructose and mannose metabolism. Organism-specific pathways can be computationally generated on the base of genes that occur in a specific organism and are participating in a particular process [KGH+06].

The group of genes can be described by the pathway in which the group members are involved. Thus biological pathways can also be used for annotation.

2.3.3 TRANSFAC

A *transcription factor* is a protein that works together with other proteins to either promote or suppress the transcription of genes.

A *binding site* is a region on a protein, DNA, or RNA to which other specific molecules and ions are bound.

A major challenge in interpreting genome sequences is understanding how the genome encodes the information that specifies when and where a gene will be expressed. The first step in this process is the identification of regions of the genome that contain regulatory information. In higher eukaryotes (organisms with complex cells: animals, plants, etc) this information is organised into modular units of a few hundred base pairs. A common feature of these modules is the presence of multiple binding sites for multiple transcription factors [BNP+02].

The TRANSFAC is the database that contains transcription factors, their binding sites, nucleotide distribution matrices and regulated genes [MKMF+06].

The database contains two types of data [MKMF+06]:

- the primary data in the database is based on experimental evidences.

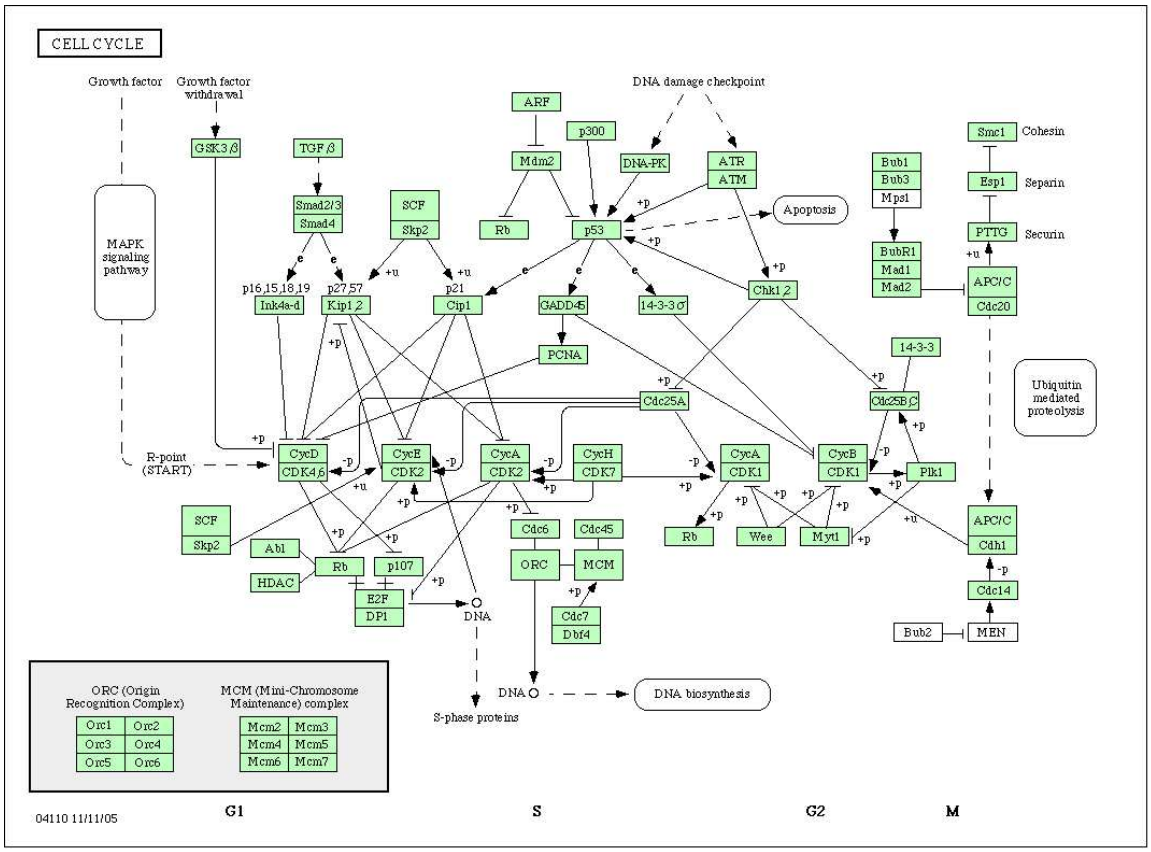


Figure 2.6: KEGG pathway hsa04110 for the life cycle of human cell. It involves 105 genes.

This kind of data is extracted by curators from scientific literature;

- the secondary data is derived from the primary data via comparison and classification.

.

The data of TRANSFAC can serve the annotation process and provide transcription factors and their bindings sites that are common for the set of genes.

Chapter 3

Visualisation and Annotation of Gene Expression Data

This chapter introduces gene expression visualisation and annotation tools. Their functionality, advantages and disadvantages are briefly discussed. At the end of the chapter we run a few steps forward and present the picture that was produced by the tool, that implements our approach to expression data analysis.

3.1 Visualisation of Large Data Sets

Hierarchical visualisation techniques are frequently used due to their capability to present information with different granularity.

The visualisation of the hierarchical clustering result consists of two parts:

- **Heatmap**, illustrating the gene expression matrix. Each dot in it represents the activity level of particular gene under particular conditions;
- **Dendrogram**, illustrating the relationships between expression profiles of genes.

Hierarchical visualisation techniques usually solve the size problem in several ways. Each of the options is illustrated by a figure. The illustrations for this chapter are produced on the same data set, which contains 882

genes from the yeast genome and their expression profiles. The data set was clustered using the agglomerative average linkage clustering and linear correlation based distance measure was used to calculate distances between items. The illustrations were obtained by different visualisation techniques:

Presentation of the complete result. The visualisation of complete data set can result in extremely large and addle picture. One might not be able to grasp the structure at a glance. This approach becomes useless with large data sets containing several thousands of items (Figure 3.1).

Node collapsing is performed so that some definite percentage of the original branches is remaining. This method enables to get the image of the desired size and is computationally cheap. But it also has crucial a drawback: the tree is cut irrespective to the patterns that can be hidden in the data set, the structure of the data is lost. It is hard to make decisions on the basis of this image (Figure 3.2).

Node collapsing is performed so that some definite number of leaves is remaining. As we can see from the Figure 3.3 it is hard to make any decision by looking at the picture. User has no information about the hidden subtrees, their density and structure. User is only provided by the size of the cluster.

The gene expression matrix can contain several tens of thousands of items. The hierarchical tree for a large data set is too big to present it in one picture. It may be difficult to comprehend the big tree. One might spend a lot of time surfing the tree on his/her own and searching for the maximally big subtrees with significant over-representation of some property.

One of the possible ways to minimise the tree is to use collapsed nodes, each of which gives the summarised view of the corresponding subtree. However, the collapsing techniques that are listed above take into account the size of the picture only. They do not take into account the underlying data nature despite of the publicly available information and knowledge. Practically they cut the tree at a particular distance, but the groups of genes with significant over-representation of the property can be lower

and higher that this cut. All the annotation and analytical tasks are left to a researcher.

Both problems, picture scalability and tree isomorphisms, need to be solved.

3.2 Gene Expression Visualisation Tools

There are several visualisation tools that build the structure of the data set. Usually, clustering can be found at the heart of these tools. The reason for this is simple. Clustering allows to perform unsupervised analysis of a big data set. It can also be performed when user has no idea about the data set itself.

3.2.1 Treeview

Treeview[1] [ESBB98, eis] is one of the most famous and eldest microarray data visualisation tools. Treeview is Windows only application that can be downloaded from Internet and installed to a computer. It provides a simple interface for viewing the results of hierarchical clustering. The clustering is done by a separate program that creates a tab-delimited text file: clustered data (CDT) file. Treeview is also used to view the results of hierarchical clustering of other types of data, including motif significance scores.

The GUI of Treeviewer is split into two parts. One part displays tree and heatmap for the whole data set. Clicking on a node in the tree will produce a zoomed image of genes included in that node. Selecting a region of the heatmap will select the smallest node containing all selected genes. This action will also produce a zoomed image. Application also allows to search the gene expression profile by a gene name.

3.2.2 Java Treeview

Java Treeview[Sal04] can be considered as an enchancement of Treeview. It is cross-platform application, thus it's functionality is available to the larger audience. It supports a generalised CDT format allowing many

[1] `http://rana.lbl.gov/`

additional details, such as colours of genes, arrays, nodes and heights of terminal branches, to be specified.

Java Treeview has several views:

- *GlobalView* displays the tree and the heatmap for the whole data set;
- *ZoomedView* displayes the subtree and the corresponding heatmap. To select a subtree user can either click on the corresponding node or drag and drop it from the tree in GlobalView;
- *GTRView, ATRView* allows to view gene trees and array trees, if the data has been hierarchically clustered using SMD[2];
- *TextView, ArrayNameView* display the available annotation for genes and arrays next to ZoomedView

3.2.3 Expression Profiler

Expression Profiler[3][Par03] is a set of web tools for microarray gene expression and other functional genomics-related data analysis. EPCLUST is one of Expression Profiler modules and is a generic data clustering, visualisation, and analysis tool for numeric (e.g. gene expression data) as well as sequence data. It allows to upload or select data sets, select parameters for clustering and cluster the data, visualise the result of clustering. One of the clustering options is hierarchical clustering. As a result of hierarchical clustering the dendrogram and tree for the whole data set is displayed. Treeview allows to cut the tree, using the "Tree Collapsing" option. The user can choose between displaying of a whole data set, specified percentage or number of branches, with the other ones collapsed into single nodes.

3.2.4 Bioconductor

Bioconductor[4] is an open source software project for the analysis and comprehension of genomic data. Bioconductor packages are distributed under the open source license, such as GPL or LGPL, and may be downloaded

[2] `http://genome-www5.stanford.edu/resources/restech.shtml`
[3] `http://www.bioinf.ebc.ee/EP/EP/`
[4] `http://www.bioconductor.org/`

from the project website for Linux, Unix, MS Windows, and Mac OS X operating systems. The initial effort was focused primarily on DNA microarray data analysis, nowadays many of the software tools, that form Bioconductor, are more general and can be used for the analysis on genomic data [Offa, Par03].

The goals of Bioconductor project are to provide access to powerful statistical and graphical methods for the analysis of genomic data, to facilitate the integration of biological metadata in the analysis.

Bioconductor allows to pre-process Affymetrix and cDNA array data, identify differentially expressed genes and plotting genomic data. It also allows to annotate data using databases such as GenBank, the Gene Ontology Consortium etc.

3.2.5 GeneSpring GX

GeneSpring GX[5] is a widespread commercial visualisation and analysis tool designed for use with gene expression data.

The latest release, GeneSpring GX 7.3, includes powerful set of analysis options. It offers a host of tools to ask questions about complex data sets: easy access to information about gene function, several linkage algorithms and plots. GeneSpring GX gives a broad choice of sophisticated methods for uncovering the most abundant patterns in the gene expression data and understanding how these patterns are related. The expression data can be displayed in several ways, e.g. pathway diagrams, classification views, 2D and 3D scatter plots.

In addition this tool has GO ontology browser, volcano plot filtering, Groovy scripting, that allows customised automated analysis [Offd].

3.3 Annotation Tools

As the name says, annotation tools are used for annotation of genes. These tools are used to provide one or a set of genes with additional information or knowledge. Hereby, several of annotation tools are briefly introduced.

[5] http://www.chem.agilent.com/

3.3.1 AmiGO

AmiGO[6] is the official tool of the GO project for searching and browsing the GO database. With AmiGO user can [Offc]:

- search for a gene or gene product, or a list of genes or gene products, and view the GO terms that are associated with the query;
- to search for pattern in the sequence and view the GO term associations for the genes or proteins returned;
- search for GO terms and view the genes or gene products they are annotated to
- browse the GO ontology and view terms.

3.3.2 g:Profiler

g:Profiler[7] is a public set of web tools for characterising and manipulating gene lists. g:Profiler has a simple, user-friendly web interface with powerful visualisation for capturing Gene Ontology (GO), pathway, or transcription factor binding site enrichments.

g:Profiler consists of several tightly integrated modules:

- g:Convert for converting between different database identifiers;
- g:Orth for finding orthologous genes from other species;
- g:Sorter for searching a large body of public gene expression data for co-expression.

g:Profiler supports 31 different species, and underlying data is updated regularly from sources like the Ensembl[8] database [RKP+07].

[6] http://amigo.geneontology.org/cgi-bin/amigo/go.cgi
[7] http://biit.cs.ut.ee/gprofiler/index.cgi
[8] http://www.ensembl.org/index.html

3.3.3 GeneTools

GeneTools[9] is a web service for gene annotation. The database of GeneTools contains annotation data from the well known, publicly available resources such as Gene Ontology. GeneTools database contains information about 64 organisms, but the most comprehensive information is available for human, rat and mouse genes. The database can be used by other tools through special protocol [BJB+06].

The web service allows user to perform:

- extraction of data for one gene or protein;
- extraction of data for batches of genes or proteins;
- manage query result lists and share selected lists with other users;
- manual GO annotation;
- export of data.

3.4 Annotation Driven Hierarchical Clustering Visualisation

As we have seen from the previous section of this chapter, both hierarchical visualisation and annotation are greatly effective and are used by researchers. The visualisation of expression data allows the researcher to track the dependencies between genes, while annotation tools can explain in what processes the requested list of genes is participating.

We propose the visualisation technique that unifies both approaches. As we have already discussed, the problem of representation scalability can be solved by collapsing of the internal nodes of the hierarchical tree. We propose **an annotation driven collapsing technique**, that takes into account the underlying data. The goal of the approach is to find the right balance between the raw data,the clustering results, and the biologically most interesting features in the data.

At the first stage the data is clustered according to its expression profiles and each hierarchical cluster is annotated. After annotation the decision

[9] `http://www.genetools.microarray.ntnu.no/common/intro.php`

of collapsing is made automatically on the basis of the list of annotations: the internal node is collapsed only if the genes, that are in the cluster have something in common, e.g. biological function. Thus only subtrees with a significant over-representation of some property are collapsed and represented by a single leaf node.

As we have stated before in Subsection 2.2.1 the tree should be uniquely determined. In order to ensure this one of the following actions is applied to each internal node of the tree:

- if both branches of internal node have collapsed nodes, then the branches are swapped so that the collapsed node that has better characteristics (bigger cluster) is the left branch of the internal node;
- if only one branch of internal node has collapsed node, then the branches are swapped so that the branch with the collapsed node is a left branch of the internal node;
- if internal node has no collapsed child-nodes, then the branches a swapped so that bigger branch is the left.

The resulting picture, produced as a result of annotation driven analysis can be found in Figure 3.4. The structure of the data set is clear, and as it is easy to see, picture guides the attention of the researcher to the significant groups of genes by itself.

The more detailed description of annotation driven hierarchical clustering visualisation technique is given in the next chapter.

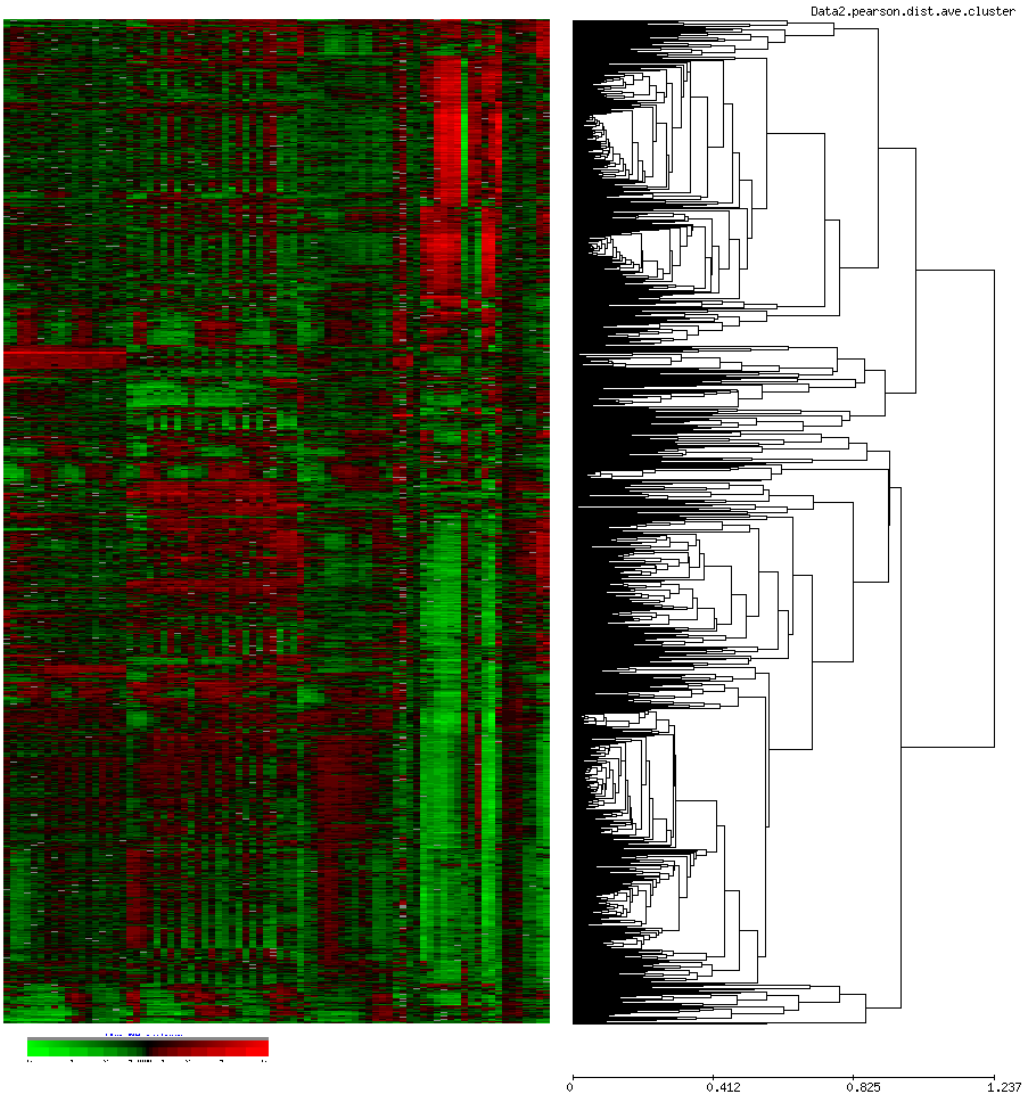


Figure 3.1: Hierarchical tree for the data set of 882 genes. It is hard to select the cluster of interest. Also, if data set would be at least two times bigger, this would be hard to fit it into the screen without loosing the details. This picture is produced by EPCLUST.

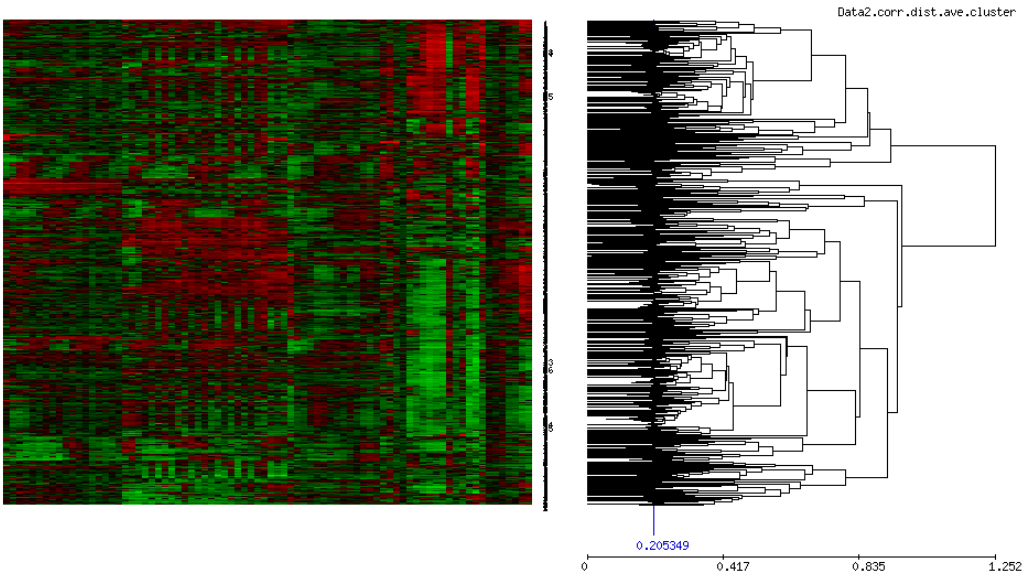


Figure 3.2: Hierarchical tree for the data set of 882 genes. From 882 objects a tree with 440 leaves (cut at distance 0.205349) is produced. Just 50% of the branches are left. Each cluster is presented as a single string in the heatmap. The eye can distinct no clusters. This picture is produced by EPCLUST.

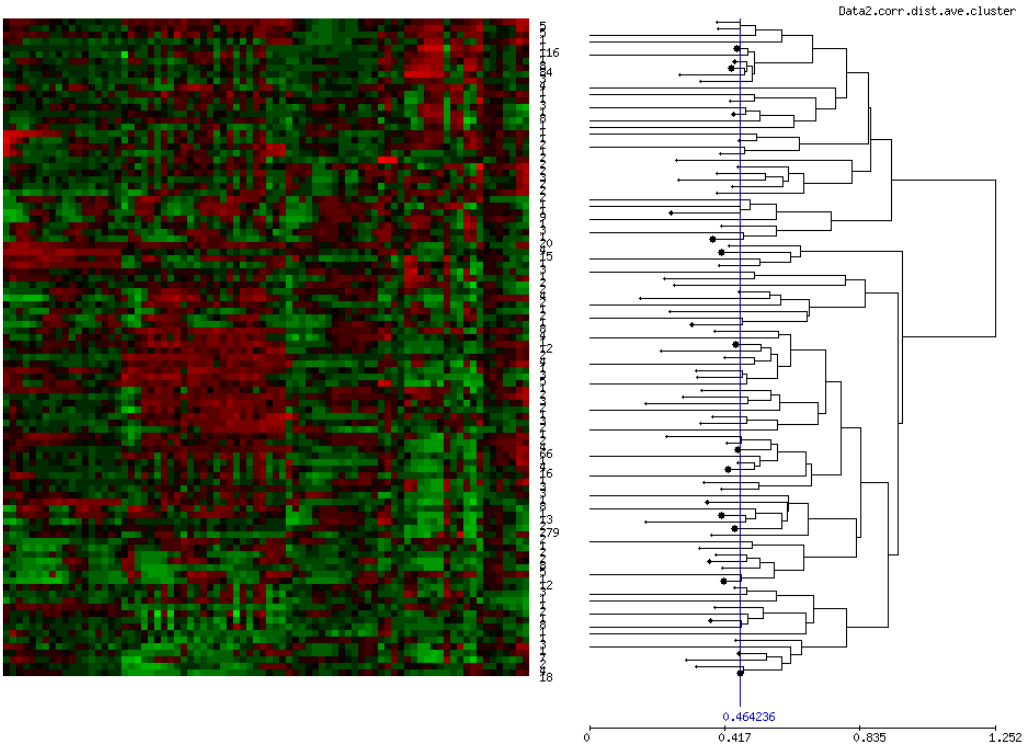


Figure 3.3: Hierarchical tree for the data set of 882 genes. From 882 objects a tree with 100 leaves (cut at distance 0.464236) is produced. Tree is cut on the basis of leaf count. The size of cluster is presented by a size of the dot in the tree, but still each cluster is presented by a single string in the heatmap. Heatmap is variegated, the user has no idea about the content of a cluster. This picture is produced by EPCLUST.

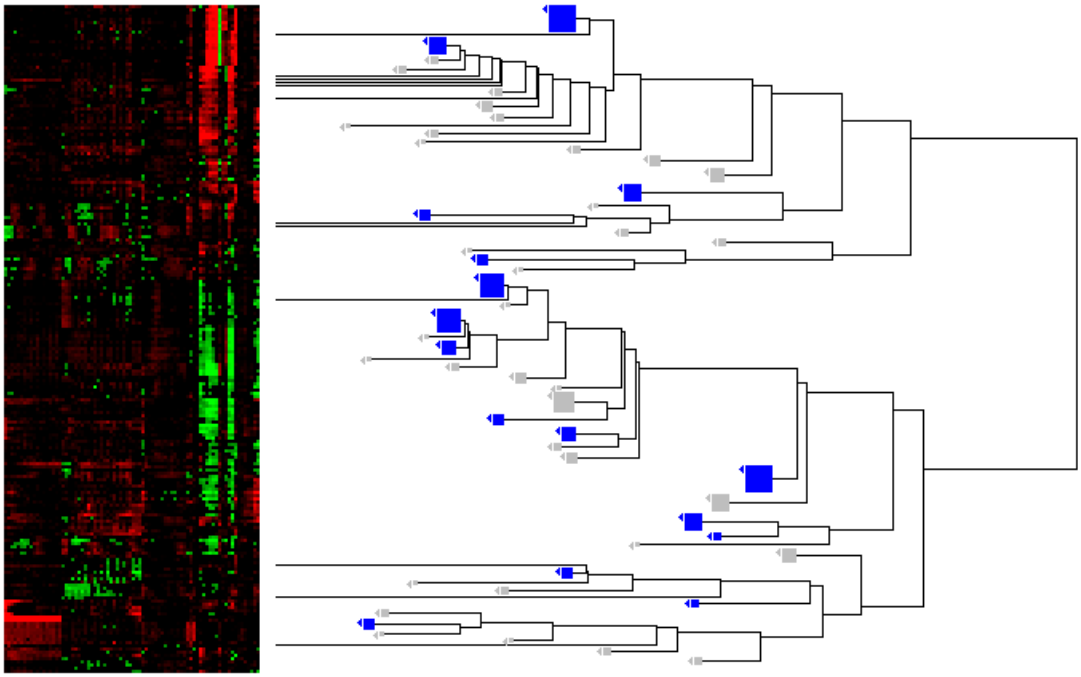


Figure 3.4: Hierarchical tree produced by the annotation driven analysis. Blue rectangles define clusters with significant over-representation of some property. Grey rectangles represent subtrees with no strong common characteristic. The size of the rectangle communicates the size of the underlying subtree.

Chapter 4

Annotation Driven Hierarchical Clustering Analysis

Current chapter introduces the annotation driven hierarchical clustering analysis that aims at finding the right balance between the raw data, the clustering results and the biologically most interesting features in the data. Hereby we introduce the hierarchical clustering analysis algorithm that makes it possible to achieve the goal.

Before we start with introducing the approach itself let us explain several definitions that are needed or specific for the current work.

Definition 1 **Cluster** is a list of leaf nodes $(l_{i,1}, ..., l_{i,j})$ that belong to a subtree s_i. Cluster has several properties:

- size $j = |(l_{i,1}, ..., l_{i,j})|$, cluster size equals to the number of leaf nodes l_i that are present in the subtree s_i;
- annotation term that expresses this group of genes the best;
- p-value ϱ_i shows the strength of the annotation term.

Definition 2 **P-value** can be defined as the probability of observing the event, summed with the probabilities of any more extremal events.

Definition 3 **Threshold** β defines the minimum strength of p-value ϱ_i. All clusters that are described by a weaker p-value can not be considered clusters that could be interesting.

Definition 4 **Dense cluster** is a cluster with a significant over-representation of some property: $\varrho_i > \beta$. An internal node n_i can not be marked as a dense cluster if it has clustered nodes among parent or descendant nodes. The size of cluster must lay within the minimum and maximum sizes, that are provided as an input. In the picture the dense cluster is presented as a single node by a blue rectangle (Figure 3.4).

Definition 5 **Thin cluster** is a cluster that does not meet the criteria of dense cluster: ϱ_i is weaker then β or the cluster is of inappropriate size. The node can not be marked as a thin cluster if it has clustered nodes among parent of descendant nodes. On the final picture the thin cluster is presented by a grey rectangle.

Definition 6 **Node collapsing process** is a process that marks the internal nodes of the tree T as a dense or a thin cluster.

4.1 Annotation Driven Hierarchical Clustering Analysis

Hierarchical clustering analysis is one of the most powerful methods for the exploratory analysis of gene-expression data. It does not need prior knowledge of the data set and provides the structure for the whole data set.

In hierarchical clustering of expression data, genes are clustered on the basis of similarity measures between expression profiles. However, apart from their position in the hierarchical tree, genes also have positional associations along the chromosomes, group of genes can be coordinated by the shared set of regulators or participate in some function. The annotation of genes can embrace Gene Ontology terms, participation in biological pathways etc.

Different tools have been developed to process and analyse gene expression data. However they have some limitations that were discussed in the previous chapter.

We propose the annotation driven hierarchical clustering analysis that tries to solve the problem of large data set visualisation by automation of hierarchical cluster analysis.

The algorithm for analysis is working on the data set where each data item is described by two sets of properties: *gene expression data* and *annotation terms*. Gene expression data is used for hierarchical clustering while annotation terms guide the node collapsing process.

The analysis expects several parameters to be specified by the researcher:

- data set D;
- the minimum and maximum possible sizes of clusters;
- threshold β that is used as a lower bound for forming dense clusters.

All parameters are mandatory.

Data set analysis consists of four stages. First of all genes are clustered according to their expression profiles. Then, each internal node of the tree is annotated. Third, dense and thin clusters are formed. Finally, the code for picture is generated and picture is presented to the researcher. See Algorithm 4.1.1 gives the overview of the algorithm.

Algorithm 4.1.1: GeneralWorkFlow

Data: data set D, $minSize$, $maxSize$, β
Result: final picture *picture*
begin
  $T \longleftarrow$ *hierarchically clustered data set D*
  $T_A \longleftarrow$ *annotated tree T*
  $T_{dense} \longleftarrow FormDenseClusters(T_A, minSize, maxSize, \beta)$
  $T_{clustered} \longleftarrow FormThinClusters(T_{dense})$
  $picture \longleftarrow createPicture(T_{clustered})$
  $display(picture)$
end

Let us consider the flow of works in more details, step by step, as it is performed.

4.1.1 Data Clustering

The genes are clustered hierarchically using their expression profiles. For instance, agglomerated average linkage clustering can be used. The hierarchical tree T represents the resulting structure of the data set.

Each leaf $l_i \in L = (l_1, l_2, ..., l_k)$ node of tree T is given a unique identifier $i \in [1..k]$. Leaf node l_i represents corresponding data item x_i from the data set X.

Internal nodes of T are also provided with identifiers: $n_{k+1}, n_{k+2}, ..., n_{2k-1}$, $|N| = k - 1$.

Subtree of internal node n_i is referred as *subtree* s_i or *cluster* s_i and represents all leaf nodes L_i that are present in the corresponding subtree. The size $|s_i|$ is equal to the number of leaves in subtree s_i.

4.1.2 Data Annotation

As far as the hierarchical tree T is ready, one might be interested in searching for "interesting" subtrees, because hierarchical clustering provides only structure that represents the relation between all objects in the data set and does not mark anything out. Here, the subtree is interesting, if there are annotation enrichments for it.

In order to find interesting subtrees we need to annotate the tree T. For each internal node n_i the group of genes belonging to the corresponding subtree s_i is created. Then, each group is annotated and its annotation is used for the corresponding internal node of the tree. If there are several annotation terms attached to the group, then the term with the strongest p-value, is selected. Thus, each internal node of the tree T is provided with one annotation term at most. This tree is referred as annotated tree T_A.

The selection of the annotation that is attached to the node is very interesting and important problem. At the moment we use one of the simplest approaches: the annotation with the best p-value is selected. However, it is also worth to consider other options: combination of top 3 annotations, combination of the most specific annotations (exclude general terms). It would be interesting to see the comparison of different selection approaches.

4.1.3 Automated Cluster Analysis

We propose node collapsing process that takes into account the biological nature of the data set and performs node collapsing process guided by the annotation terms of internal nodes of T. The collapsing algorithm searches for the dense and thin clusters in the tree structure. The threshold β, possible minimum and maximum sizes of cluster should be provided as an input.

The node collapsing is performed in two steps. First of all this process searches for the dense clusters. For this the list containing all internal nodes of T is built and sorted descending by the p-values of annotations. While iterating over the list the collapsing process tries to form dense clusters. If parameters of the cluster satisfy the conditions of (1)threshold β, (2)size and (3)node has no clustered parent or descendant nodes, it is marked as dense cluster.

At the second stage the collapsing process tries to minimise the tree T as much as possible by collapsing internal nodes that can be turned to thin clusters. The process passes the tree T recursively starting from the root and searches nodes that meet thin cluster criteria. If the node satisfies the thin cluster criteria, then it is marked so. Process continues its work and takes another branch of the tree and scans further, until all branches are traversed.

The algorithm used for the node collapsing can be found in Algorithm 4.1.2 and Algorithm 4.1.3.

Algorithm 4.1.2: FormDenseClusters

```
Data: annotated tree T_A, minSize, maxSize, threshold β
Result: T_dense with formed dense clusters.
begin
    nodes ⟵ list of n ∈ T_A
    sort nodes by ϱ, desc
    while node ∈ nodes and node.pValue ≥ β do
        if node.hasNoCollapsedDescendants() and
        node.hasNoCollapsedParents() and node.size ≥ minSize and
        maxSize ≥ node.size then
            collapseDenseCluster(node)
end
```

Algorithm 4.1.3: FormThinClusters

```
Data: s is the subtree of T_dense
Result: T_clust
begin
    if node.hasChildren() and node.isNotClustered() and
    node.hasNoCollapsedDescendants() then
        collapseThinCluster(node)
    else if node.hasChildren() then
        FormThinClusters(node.leftChild)
        FormThinClusters(node.rightChild)
end
```

4.1.4 Data Visualisation

Finally, when the tree is built, annotated and collapsed, the data needs to be presented to the user in such a way, that he/she would be able to gain the understanding of it.

In bioinformatics gene expression data is often presented by the heatmap and thedendrogram, connected to each other.

Displaying the raw data set, that might consist of 6,000 items, (this is one of the possible sizes for the data set for baker yeast(*lat. Saccharomyces cerevisiae*)) would take too much vertical space and processing time. The data sets for mouse or human genome are 4 – 5 times larger. This would make this kind of visualisation hardly understandable. It is impossible to grasp it at once.

The techniques used by the hierarchical clustering visualisation algorithm presents the data with a heatmap and dendrogram, that were created using T_{clust}. The nodes that were clustered during node collapsing stage are represented by rectangles. To keep the image of a small size and to pass the information concerning the size of the cluster, the size of the rectangle is calculated by scaling the size of the cluster. The coefficients for the formulae were tuned empirically. As the result the scaling is performed using Formula 4.1 and Formula 4.2.

$$scaledSize = \left\lceil 3.5 * size^{\frac{1}{3}} \right\rceil \tag{4.1}$$

$$scaledSize = \begin{cases} 50 & \text{if } scaledSize \geq 50, \\ 4 & \text{if } scaledSize \leq 4, \\ scaledSize & \text{otherwise.} \end{cases} \tag{4.2}$$

The comparison of usage of other coefficients can be found in Appendix A.

Dense clusters are represented by blue rectangles and thin clusters — by grey ones. The size of the rectangle provides the idea of the size of the underlying cluster. The rows of the heatmap that correspond to the clusters contain random gene profiles from the appropriate cluster, thus giving an idea of the whole cluster (Figure 4.1).

Used visualisation techniques are intuitive for biologists and based on the traditional expression data view with the dendrogram. However, subtrees with a significant over-representation of some property are collapsed and represented by a single leaf node. This gives a good overview of the data while pointing out different regions having enrichment in different properties. Collapsed nodes allow to achieve the wanted size of the data set. The technique of collapsing allows to express the biological nature of the data set to the user. In this case hierarchical clustering visualisation gives the exploratory analysis of the data set bringing to the forefront the groups of genes that biologically significant and hiding the groups of genes with poor biological background.

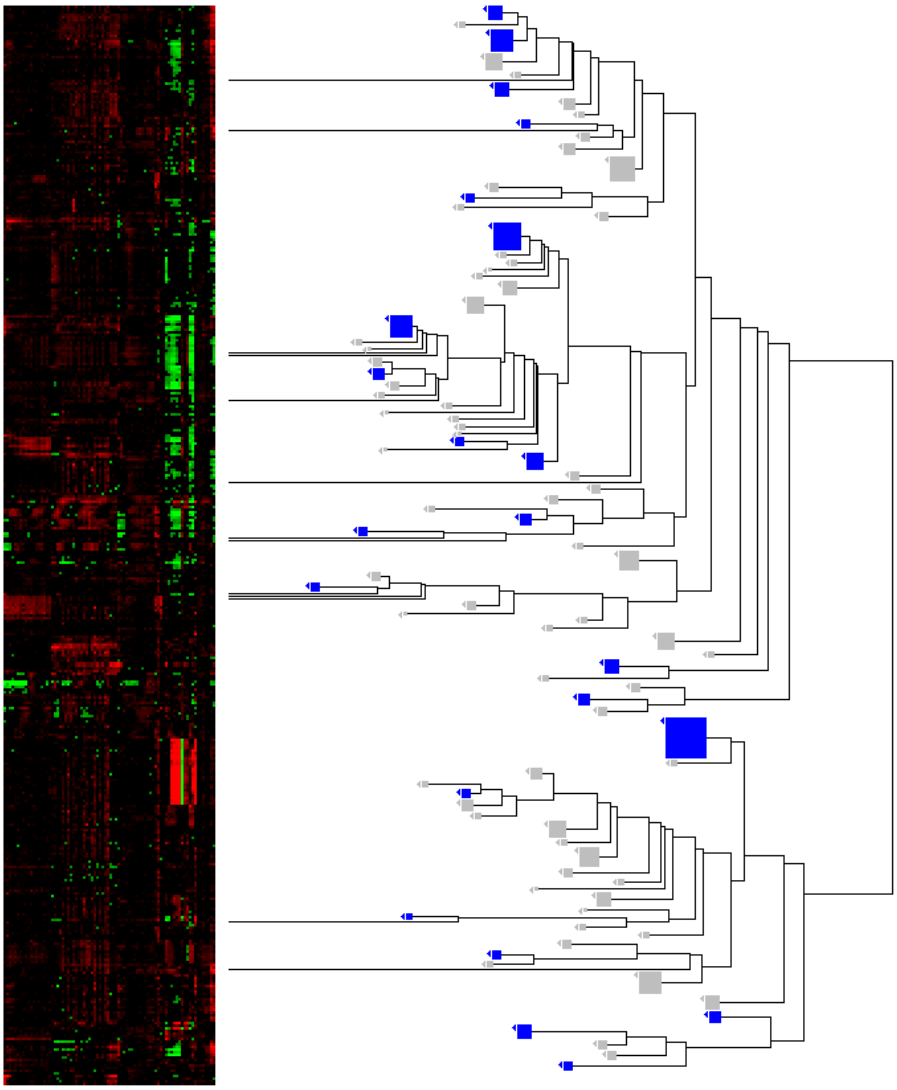


Figure 4.1: Hierarchical tree produced by the annotation driven analysis. Blue rectangles define clusters with significant over-representation of some property. Grey rectangles represent subtrees with no strong common characteristic. The size of the rectangle communicates the size of the underlying subtree.

Chapter 5

Applications Used for Annotation Driven Analysis

Previous chapters contained an introduction into problem and the main goal of the work. This chapter contains description of tools that are used during the annotation driven analysis.

5.1 Happieclust

There are several clustering algorithms, but they face the problem of efficiency. The common problem with the k-medoids and hierarchical clustering methods is that they calculate $\Omega(n^2)$ distances between the data items, where n is the number of items in the data set. This could be a bottleneck for expression data analysis and visualisation tool. In addition the efficiency is highly dependant on the amount of data.

In his master thesis Meelis Kull proposed fast approximate hierarchical clustering [Kul04]. This approach overcomes the problem of efficiency and proposes complexity of $O(n * \log n * \log m + m)$, where m is the number of pairwise distances to be calculated, and n is the number of vertices in the data set. Due to the size of expression data a fast approximate clustering algorithm is an option.

Due to the fact that only fraction of all pairwise distances are calculated, the improvement of complexity is achieved. This approach resulted in the poor quality of clusters. But it also appeared that by careful selection of the pairs for which the distances are calculated the quality of clusters can

be improved.

A special method was developed to find pairs of similar items. This method can be used to find all pairs that are more similar to each other than some threshold value.

The experiments showed that the approximate hierarchical algorithm together with the best strategy for choosing pairs works in almost linear time in the number of calculated distances.

This approach still has some open questions, but it is more effective and produces accurate result [Kul04].

Happieclust[1] is the implementation of fast approximate hierarchical clustering and is used to cluster genes by their expression patterns. The detailed explanation of clustering algorithm can be found in [Kul04].

5.2 g:Profiler

g:Profiler is a web service that is used for annotation of gene groups. The list of annotation, containing annotations for each subtree in hierarchical tree is used to guide node collapsing process.

The genes and groups of genes can be annotated using GO terms, KEGG pathways etc. g:Profiler [gpr, RKP+07] makes it possible to annotate a list of gene groups. It can be considered as separate annotation of each group of genes.

The annotation of single group of genes, or *query*, can be described as following:

During the first stage of annotation process g:Profiler searches for all annotation terms that are related to at least one gene from the query.

At the next step the set of the annotations is improved by filtering out statistically insignificant annotations.

Finally, the terms that are left are grouped into a graph, using the relations between the terms. The parent and descendant nodes of the graphs that are too general to be interesting or too detailed to characterise the data set as a whole are discarded by the significance filter. The terms that are left can be considered as terms that describe the given group of genes.

[1] `http://biit.cs.ut.ee/book-page/2007/01/08/approximate-hierarchical-clustering`

One of the most important steps during the annotation of group of genes is the filtering of statistically insignificant annotations. It is done by means of a hypergeometric p-value which is calculated for each annotation.

The hypergeometric probability can be described by the classical urn problem as following: *Let urn contain N balls: K white and N-K black balls. Let us pick without replacement n balls. What is the probability of getting exactly k white balls and n-k black balls?*

Hypergeometric probability is calculated with following formula:

$$p_h(n,k;N,K) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}};$$

In case of evaluating common elements for a gene query G_q and an annotation set A, $N = |G|$ is the total number of the genes, and $K = |A|$ is the number of genes annotated to the term $a \in A$. The value $n = |G_q|$ is the number of genes in the query, and $k = |G \cap A|$ is the number of genes in intersection of $G \cap A$. In this case above formula cam be rewritten with following parameters:

$$p_h(G \cap A) = p_h(|G_q|, |G_q \cap A|; |G|, |A|);$$

Exact calculation $G_q \cap A$ is not sufficient according to the definition of p-value. In order to assess significance of match, we need to consider the probability of exact arrangement, as well as all possible more extreme arrangements:

$$p_{ch}(k \geq x) = p_{ch}(n,k;N,K) = \sum_{k=x}^{n} \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}};$$

More detailed explanation of the p-value calculation and algorithm reader can find in [Rei06, RKP+07].

5.3 Data Limitations

The annotation driven clustering analysis works on data sets where each object is described by two sets of properties. The sets are used separately at different stages of the algorithm — the first set is used for hierarchical

clustering whereas the other set, which consists of binary features only, is used for node collapsing.

Current work considers the application of the algorithm to the gene expression data, but it also can be applied to the other types of data, that satisfy the conditions.

To achieve the goal annotation driven analysis integrates implementations of fast approximate hierarchical clustering and g:Profiler annotations. The output of these tools is used as an input for the analysis.

Different data sets are used to perform the analysis. Altogether 4 data sets ate used.

Gene Expression Data Set Data sets of this type are used by clustering software and contain gene expression data. As we described before, gene expression data is a numerical matrix where the rows represent the genes and the columns represent the experiments. Each intersection of row and column holds the expression value of a single position within the microarray.

The gene and experiment names are omitted from the data set. The order of rows and columns is respected, so that the names can be derived later, when needed.

This data set is also used to display a heatmap.

Gene Names Data Set Set with gene names is used to identify the nodes of the hierarchical tree by corresponding gene names. The content of the data set are gene names, each on the separate row. The order of elements in the set is preserved.

Two other types of data sets are obtained on the basis of previous two.

Hierarchical Clustering Data Set The data set is produced by Happieclust on the basis of corresponding gene expression data set and contains description of the hierarchical tree. Tree is built according to the description contained in the hierarchical clustering data set. The example of data set is shown in Figure 5.1.

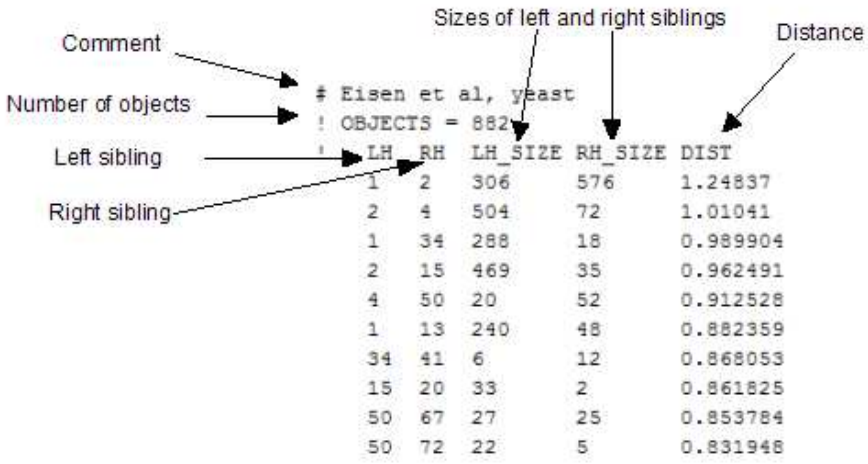


Figure 5.1: Hierarchical clustering data set contains header and description of the tree. The description itself consists of 5 columns: id of left sibling, id of right sibling, size of left sibling, size of right sibling and distance between two clusters.

1	2	3	4	5	6	7	8	9	10	11	12
1	!	1.29e-05	24	2	2	1.000	0.083	GO:0010035	BP	2	response to inorganic substance
1	!	7.17e-06	18	2	2	1.000	0.111	GO:0010038	BP	2	response to metal ion
1	!	4.69e-07	5	2	2	1.000	0.400	GO:0046688	BP	2	response to copper ion
1	!	1.65e-05	27	2	2	1.000	0.074	GO:0005507	MF	1	copper ion binding
3	!	1.30e-04	75	2	2	1.000	0.027	GO:0016052	BP	7	carbohydrate catabolic process
3	!	1.30e-04	75	2	2	1.000	0.027	GO:0044275	BP	7	cellular carbohydrate catabolic process
3	!	3.12e-05	37	2	2	1.000	0.054	GO:0006090	BP	7	pyruvate metabolic process
3	!	3.12e-05	37	2	2	1.000	0.054	GO:0046165	BP	7	alcohol biosynthetic process

Figure 5.2: Annotation data set contains query identifier(1), significance (exclamation point) or insignificance of the annotation (no exclamation point) of the annotation(2) p-value(3), number of genes associated with term (4), the size of a query(5), size of intersection of genes in the query with annotations(6), precision(7), recall(8), annotation term id(9), code of domain(10), number of the graph where annotation is located(11), name of term(12)

Annotation Data Set Annotation data set is built by g:Profiler based on list of clusters derived from the hierarchical tree. The list of groups is build according to the internal nodes of the hierarchical tree. The sample and the description of the data can be found in Figure 5.2.

Chapter 6

The Treeviewer Tool

In the previous chapter we have introduced the approach of hierarchical clustering visualisation. This technique takes into account the nature of the data.

The goal of this chapter is to introduce a tool, that was created as a part of the master thesis to demonstrate the approach in action. We will give the short overview of an architecture and detailed description of possible usage scenarios, implemented at the moment.

Examples of the graphical user interface can be found in Appendix B. The latest version of the tool can be found at `http://emu.at.mt.ut.ee/treeviewer/`.

6.1 Architecture

Treeviewer[1] is a JEE[2] web application for performing hierarchical clustering visualisation for the gene expression data. It is platform independent and works in any servlet container or application server, such as Tomcat[3] or JBoss[4]. Since it is a web application, its services are accessible for everyone who has access to the Internet.

The application itself is broken into several tiers (Figure 6.1):

[1] `http://emu.at.mt.ut.ee/treeviewer/`
[2] `http://java.sun.com/javaee/technologies/javaee5.jsp`
[3] `http://tomcat.apache.org/`
[4] `http://labs.jboss.com/portal/`

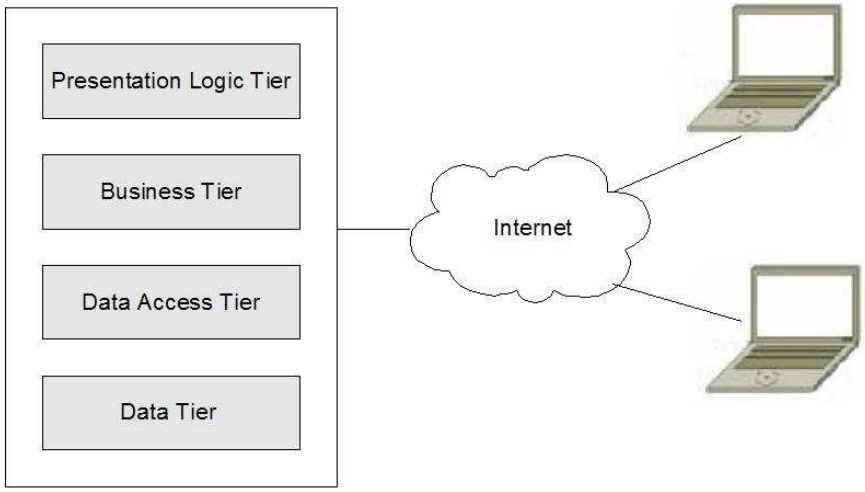


Figure 6.1: Treeviewer architecture.

Data Tier. It is responsible for storage of initial and precalculated data sets.

Current version of Treeviewer supports 2 types of data tiers: local file system and GED.

GED is Gene Expression Database tool that was developed inside BIIT[5] research group. The aim of the module is to simplify expression data management and integrate various tools, developed inside the group. Within GED, the data sets are stored in the files. GED allows to upload and retrieve data sets. It provides special web based interface for Treeviewer: the plain text, tab separated data sets can be queried by the data set type (e.g. clustering or annotation) and data set name.

Data Access Tier. It contains generic methods for querying the data from the data tier. This tier abstracts the usage of the data tier and permits fast and simple switching between various implementations of the latter.

Business Tier. This layer contains code that is responsible for the business logic of the application: building, annotating and collapsing of trees.

[5] `http://biit.cs.ut.ee/`

The Presentation Logic Tier. It is implemented using Tapestry framework[6]. The tier contains components and html templates that are used to build dynamic HTML pages. Also the SWOG library[7] is used in this tier in order to generate images for hierarchical trees and subtrees.

Presentation GUI. The application has got web user interface and can be accessed using any up-to-date web browser.

Spring Framework[8] is used as a backbone of the application. It decoples implementations of different tiers and improves testability of the code.

The application building and deploying is made using software project management and comprehension tool Maven[9].

6.2 Implementation of Analysis

As we have mentioned in the previous chapter Treeviewer is using two precalculated data sets: hierarchical clustering data set produced by Happieclust and annotation dataset produced by g:Profiler. The main reason for this is reusability of data sets. Besides clustering and annotation are time consuming procedures.

The implementation of the algorithm can be briefly described as following.

The analysis uses as an input 4 data sets (gene expression data set, gene names data set, hierarchical clustering data set and annotation data set) and 3 parameters (minimum and maximum possible sizes for the dense cluster and threshold β).

First of all, precalculated hierarchical clustering is queried from the data tier and hierarchical tree T is built according to it. Then the list of annotations A for the data set under analysis is loaded. It may contain several annotations for the single internal node, at the same time some

[6] `http://tapestry.apache.org/`

[7] `http://www.bioinf.ebc.ee/~hansen/swog/` [Han05]

[8] `http://www.springframework.org/`

[9] `http://maven.apache.org/`

nodes may have no annotation terms at all. During the iteration of annotation in the list A we try to provide each internal node of tree T with an annotation that describes corresponding subtree in the best possible way. The algorithm that describes procedure of the tree annotation can be found in Algorithm 6.2.1.

Algorithm 6.2.1: AnnotateTree

Data: hierahchical tree T, list of annotations A, sorted by p-value, desc
Result: annotated hierarchical tree T_A
begin
  for $a_i \in A$ **do**
    if n_i *has no annotation* **or** $n_i.annotation.pValue \geq a_i.pValue$ **then**
      $setAnnotation(n_i, a_i)$
end

As far as annotation of the tree is ready, the collapsing process forms dense and thin clusters as it is described in Algorithm 4.1.2 and Algorithm 4.1.3. Finally pictures that illustrate the clustered tree are generated.

6.3 Usage Scenarios

Lets cover usage scenarios that are implemented at the moment.

6.3.1 Single Data Set Analysis

The most basic scenarios for the data set analysis is a single data set visualisation. This option gives the overview of selected data set by creating picture with dendrogram and heatmap and list of significant clusters.

User is also able to search for a gene or genes by inserting the list of gene names and submitting the form.

As a result of the analysis, the picture presented to the user. It is a clickable image, that allows the user:

- consider the analysis of the dataset (Figures 3.4 and Figure 4.1);
- to open and close clusters (Figure B.4);

- select cluster and consider it separately (Figure B.6);
- to see the location of searched genes (Figure B.5).

6.3.2 Single Cluster Analysis

By selecting a single cluster from the dendrogram for the whole data set, the user is guided to the single cluster analysis page (Figure B.6).

Here the user can see the cluster members and annotation term that is over-represented in the cluster. The dendrogram of subtree contains no collapsed nodes anymore, thus each gene is displayed.

From this page the user can move on to the other analytical tools and

- search for other annotation terms for this group of genes using g:Profiler;
- convert the names of the genes from the group;
- look for external resources;
- search for the genes in different species, that are similar to the genes from query and originated from a common ancestor.

6.3.3 Data Set Comparison

Data comparison analysis differ from both of the previous analysis. It allows to compare dense clusters of two or more data sets.

For instance, the user has got two data sets: one for the healthy and another is for a diseased organism. One may surmise, that the genes in healthy and diseased organism are working in a different way, but it would be interesting to understand what is changed: which terms are not significant anymore? which, otherwise, became significant? and have the p-value of dense clusters changed or not?

This kind of analysis visualises the differences in data sets, and helps to emphasise the changes in the behaviour.

To perform this kind of analysis the user starts with selection of the data sets that he/she wants to compare from the list of available data sets. After that the calculations are performed and the comparison table is presented. Each row of the table correspondss to the annotation term

that characterises at least one of the dense clusters of the compared data sets. Each column represents one of the compared data sets. The cell of the table contains p-value and the size of the cluster (for particular data set and particular cluster, that is described by the term). If one of the data sets has no cluster with some particular term, then corresponding cell is left empty (Figure B.8).

This tabular representation of data helps to follow the dynamics of the genes behaviour.

Chapter 7

Conclusions

Advanced technologies have made it possible to track and store large amounts of multidimensional data. The fast development of technology has also affected the course of the research in molecular biology. Nowadays, microarray experiments are conducted to learn more about cellular states. Each experiment may contain 40,000+ measurements that characterise the behaviour of different genes under different conditions.

Hierarchical clustering is often used to perform exploratory analysis on gene expression data. Even though a clustering algorithm is usually able to output some structured data, the textual output is rarely used. However, even a picture of a tree that contains several thousands of genes turns out to be too big to be properly understood by a human. To minimise the tree, the collapsed nodes have to be used, each of which gives a summarised representation of the collapsed subtree. Usually, collapsing is performed on the subtrees that are either at a fixed depth or of a fixed size.

The main goal of the thesis was to propose and implement a new algorithm that would aim at finding the right balance between the raw data, the clustering results, and the biologically most interesting features in the data. The algorithm operates on data sets where each object is described by two sets of properties. The sets are used separately at different stages of the algorithm. The first set is used for hierarchical clustering while the other set is used for guiding the node collapsing process. The algorithm looks for subtrees with significant over-representation of some binary feature and favours these in the collapsing process. This annotation driven visualisation not only minimises the picture, but also performs the automated analysis of the tree and highlights the subtrees of particular

interest.

We have also built a microarray gene expression data visualisation web tool which implements this technique. The hierarchical tree is derived by a fast hierarchical clustering algorithm [Kul04] and the node collapsing process is guided by the list of annotations that is obtained from g:Profiler [RKP+07]. The automated analysis is performed to minimise the tree and highlight the groups of genes with over-represented features. Finally, the resulting navigable picture of the tree is presented to the user.

The pictures created using the proposed technique are lightweight and can be grasped at a glance. However, they still communicate the structure of the data set and emphasise the clusters with significant features.

Our first tests have shown that this algorithm proposes much more than just the visualisation of a single data set. It enables many-sided cross- and inner- species analyses and the comparison of results. We plan to extend the analytical and the visualisation features of Treeviewer to enable a more diverse analysis of gene expression data. For instance, a more sophisticated method for the selection of the annotation that guides clustering may produce better results of finding the gene groups with over-representation of some features.

The latest version of Treeviewer application is located at `http://emu.at.mt.ut.ee/treeviewer/`.

Annotatsioonidel põhinev hierarhilise klastrdamise analüüs

Magistritöö (40AP)

Darja Kruševskaja

Sisukokkuvõte

Uuemad tehnoloogiad võimaldavad suurte hulkade multidimensionaalsete andmete tootmist ning salvestamist. Tehnoloogia areng on mõjutanud ka molekulaarbioloogia uurimustööde suunda ning kiirust. Käesoleval hetkel on juba sekveneeritud mitmete imetajate genoomid. Teadlased otsivad vastust küsimusele, kuidas genoom kui tervik funktsioneerib?

Tänapäeval on teada, et genoom koosneb funktsionaalsetest ja mittefunktsionaalsetest sekventsidest. Geeni ehk funktsionaase järjestuse tuvastamine on üks esimesi ja tähtsamaid samme, mis genoomi funktsioneerimisest paremini aru saamiseks. Samas on oluline eristada funktsionaalse järjestuse tuvastamist järjestuse funktsiooni leidmisest. Geeni konkreetset funktsiooni kindlaks määramine on omaette ülesanne.

Varasematel aegadel oli vaja geeni funktsiooni määramiseks korraldada spetsiaalseid *in vivo* katseid. Kui funktsioon oli kindlaks tehtud ja eksperimentaalselt kinnitatud, siis publitseeriti vastav teadusartikkel. Selline lähenemine oli ebaefektiivne: kallis, aeglane, vaevaline, teadmised olid hajutatud ja ühisterminoloogia puudus.

Tänapäeval on võimalik teostada mikrokiipidega mahukaid katseid ning mõõdetakse üheaegselt paljude geenide ekspressiooni. Selle tulemusena, on olemas mahukad geeni ekspressiooni andmestikud. Neid andmestikud sisaldavad andmeid geenide käitumisest.

Ühisterminoloogija puudumine oli pikka aega genoomi uuringutes pidurdavaks faktoriks. Olemasoleva informatsiooni struktureerimiseks ja süstematiseerimiseks käivitati mitmeid projekte (näit. *Geeniontoloogia*) Tekkisid ka avalikud terminite. Neid sisaldavad termineid, millega saab iseloomustaga geene ja geenide produkte.

Tavaliselt algab geeni ekspressiooni analüüs klasterdamisega: geenid klasterdatakse ekspressiooni profiilide järgi: sarnase käitumisega geenid pannakse samasse gruppi ja erineva käitumisega erinevatesse gruppidesse. See võimaldab uurijale edaspidi keskenduda ühele, mingite näitajate poolest huvitavale, grupile ja seda siis põhjalikumalt vaadelda.

Paljud tööristad kasutavad ekspressiooniandmete analüüsiks hierarhilist klasterdamist ja esitavad tulemuse interaktiivse pildina. Tulemuse visualiseerimine on analüüsi ülioluline osa, sest puu võib sisaldada palju tippe (inimgenoomi puhul umbes 30 000 tippu). Tekstifailist on väga raske ammutada sellise suurusega andmestiku struktuuri, isegi pilt võib osutuda liiga suureks ja kirjuks.

Selleks, et pildil olevate elementide arvu vähendada võib mõned tipud kokku võtta. Selline kokkuvõetud tipp esindab temast lähtuvat kogu alampuud. Tavaliselt võetakse tippe kokku kas puu kindlal sügavusel või rakendatakse seda protseduuri kindla suurusega alampuudele. Selline lähenemine ei too samuti esile kogu andmestiku struktuuri ja andmete lõpliku analüüsi peab teostama uurija.

Meie pakkume käesolevas magistritöös välja annotatsioonidel põhineva hierarhilise klasterdamise analüüsi. Sellise analüüsiga saadav tulemus toob esile andmestiku struktuuri kompaktsel kujul ja juhib tähelepanu alampuudele, mis sisaldavad statistiliselt olulisi annotatsioone.

Selle uurimistöö tulemuseks on algoritm, mis püüab saavutada tasakaalu algandmete, klasterdamise tulemuste ning andmete bioloogiliselt huvitavate tunnuste vahel. Puud minimaliseeritakse siin annotatsioonidest lähtudes.

Valmis on ka algoritmi esimene realisatsioon Treeviewer. See klasterdamiseks ja annoteerimiseks ühendab kaks juba olemasolevat rakendust: Happieclust [Kul04] ning g:Profiler [RKP+07]. Käesolev magistritöö keskendub automaatsele puu analüüsile, puu minimiseerimisele ning visualiseerimisele.

Loodud rakenduse viimane versioon asub aadressil `http://emu.at.mt.ut.ee/treeviewer/`.

Acknowledgements

The last page of the thesis I would like to dedicate to all people who supported me during the master studies.

Above all, I would like to thank my supervisor Dr. Jaak Vilo for guiding me during last three years. His motivating ideas and cognitive conversations made this thesis real. His hard-working and devotion infects people around him.

I am also gratified to Jaak for introducing me into his research group BIIT. I have found there new friends, who helped me to survive through the period of studies. Their ideas, feedback and help were essential. Especially I would like to mention Meelis Kull and Jüri Reimand for their ideas and help with the implementation of Treeviewer application. I want to thank Hedi Peterson for guiding me through the shadows of biology, and Dr. Maarika Traat for participating in the final circle of thesis review. Raivo, Kostja, Ilja, Jaanus, Pavlos, thank You all for your support.

I also would like to thank Liina Kamm and Dan Bogdanov for being great companions during this exhausting period.

I want to thank my family for being supportive and ready to help.

Last, but not least I would also like to thank Anton Litvinenko for being my conscience. I would like to thank him for his patience, support and helping out with design issues of Treeviewer application.

This Master's Thesis has been partially supported by ETF grants (ETF-5722 and ETF-5724), as well as ENFIN project (LSHG-CT-2005-518254).

Bibliography

[ABB+00] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[ACH+00] M.D. Adams, S.E. Celniker, R.A. Holt, C.A. Evans, J.D. Gocayne, P.G. Amanatides, S.E. Scherer, P.W. Li, R.A. Hoskins, R.F. Galle, et al. The Genome Sequence of Drosophila melanogaster. *Science*, 287(5461):2185–2195, 2000.

[BJB+06] V. Beisvag, F.K.R. Junge, H. Bergum, L. Jolsum, S. Lydersen, C.C. Gunther, H. Ramampiaro, M. Langaas, A.K. Sandvik, and A. Laegreid. GeneTools–application for functional annotation and statistical hypothesis testing. *BMC Bioinformatics*, 7(1):470, 2006.

[BNP+02] B.P. Berman, Y. Nibu, B.D. Pfeiffer, P. Tomancak, S.E. Celniker, M. Levine, G.M. Rubin, and M.B. Eisen. Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. *Proceedings of the National Academy of Sciences*, 99(2):757–762, 2002.

[BV00] A. Brazma and J. Vilo. Gene expression data analysis. *FEBS Lett*, 480(1):17–24, 2000.

[CMB+04] E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez, R. Apweiler, et al. The Gene Ontology Annotation (GOA) Database: sharing

knowledge in Uniprot with Gene Ontology. *Nucleic Acids Research*, 32(90001):W313–W317, 2004.

[CVS+05] Z. Cheng, M. Ventura, X. She, P. Khaitovich, T. Graves, K. Osoegawa, D. Church, P. DeJong, R.K. Wilson, S. Pääbo, et al. A genome-wide comparison of recent chimpanzee and human segmental duplications. *Nature*, 437(7055):88–93, 2005.

[DTW05] R.C. Deonier, S. Tavaré, and M.S. Waterman. *Computational Genome Analysis: An Introduction*. Springer, 2005.

[eis] Eisen lab. `http://rana.lbl.gov/`. [Online; accessed 12-May-2007].

[ESBB98] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns, 1998.

[GBB+96] A. Goffeau, BG Barrell, H. Bussey, RW Davis, B. Dujon, H. Feldmann, F. Galibert, JD Hoheisel, C. Jacq, M. Johnston, et al. Life with 6000 Genes. *Science*, 274(5287):546–567, 1996.

[gpr] g:profiler. `http://www.bioinf.ebc.ee/gprofiler/`. [Online; accessed 12-May-2007].

[GSvH00] DR Gilbert, M. Schroeder, and J. van Helden. Interactive visualization and exploration of relationships between biological objects. *Trends Biotechnol*, 18(12):487–94, 2000.

[Han05] J. Hansen. Graphics language SWOG. *Bachelor thesis, University of Tartu*, 2005.

[HMB+04] L.D.W. Hillier, W. Miller, E. Birney, W. Warren, R.C. Hardison, C.P. Ponting, P. Bork, D.W. Burt, M.A.M. Groenen, M.E. Delany, et al. Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature Publishing Group*, 432(7018):695–716, 2004.

[JMF99] AK Jain, MN Murty, and PJ Flynn. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 1999.

[JTZ04] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: a survey. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1370–1386, 2004.

[Kar01] P.D. Karp. Pathway Databases: A Case Study in Computational Symbolic Theories. *Science*, 293:2040–2044, 2001.

[KBH+03] E.F. Kirkness, V. Bafna, A.L. Halpern, S. Levy, K. Remington, D.B. Rusch, A.L. Delcher, M. Pop, W. Wang, C.M. Fraser, et al. The Dog Genome: Survey Sequencing and Comparative Analysis, 2003.

[KGH+06] M. Kanehisa, S. Goto, M. Hattori, K.F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, M. Hirakawa, and O. Journals. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34:D354–D357, 2006.

[Kul04] M. Kull. Fast clustering in metric spaces. Master's thesis, University of Tartu, 2004.

[Lew05] S.E. Lewis. Gene Ontology: looking backwards and forwards. *Genome Biology*, 6(1):103, 2005.

[LLB+01] E.S. Lander, L.M. Linton, B. Birren, C. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[MKMF+06] V. Matys, OV Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, et al. TRANSFAC® and its module TRANSCompel®: transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34:D108–D110, 2006.

[MS99] C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[Offa] Bioconductor: Open source software for bioinformatics. `http://www.bioconductor.org/`. [Online; accessed 12-May-2007].

[Offb] European bioinformatics institute. http://www.ebi.ac.uk/. [Online; accessed 12-May-2007].

[Offc] The gene ontology project. http://geneontology.org/. [Online; accessed 12-May-2007].

[Offd] Genespring gx. http://www.chem.agilent.com/scripts/pds.asp?lpage=27881/. [Online; accessed 12-May-2007].

[Offe] Kegg: Kyoto encyclopedia of genes and genomes. http://www.genome.jp/kegg/. [Online; accessed 12-May-2007].

[Offf] National human genome research institute. http://www.genome.gov/. [Online; accessed 12-May-2007].

[Par03] G.G. Parmigiani. *The Analysis of Gene Expression Data: Methods and Software*. Springer, 2003.

[Rei06] J. Reimand. Gene ontology mining tool gost. Master's thesis, University of Tartu, 2006.

[RKP+07] J. Reimand, M. Kull, H. Peterson, J. Hansen, and J Vilo. g:Profiler – a web-based toolset for functional profiling of gene lists from large-scale experiments. http://nar.oxfordjournals.org/cgi/content/abstract/gkm226?ijkey=HfWHjpSUFSJiuJV&keytype=ref, 2007.

[Sal04] A.J. Saldanha. Java Treeview – extensible visualization of microarray data. *Bioinformatics*, 20(17):3246–3248, 2004.

[Sch04] C.F. Schaefer. Pathway Databases. *Annals of the New York Academy of Sciences*, 1020(1):77–91, 2004.

[SES01] R. Sharan, R. Elkon, and R. Shamir. Cluster analysis and its applications to gene expression data. *Ernst Schering workshop on Bioinformatics and Genome Analysis*, 2001.

[SND05] P. Saraiya, C. North, and K. Duca. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Information Visualization*, 4(3):191–205, 2005.

[TSS06] A. Tanay, R. Sharan, and R. Shamir. *Handbook of Computational Molecular Biology*. CRC Press, 2006.

[WLTB+02] R.H. Waterston, K. Lindblad-Toh, E. Birney, J. Rogers, J.F. Abril, P. Agarwal, R. Agarwala, R. Ainscough, M. Alexandersson, P. An, et al. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420(6915):520–562, 2002.

[YHR01] KY Yeung, DR Haynor, and WL Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.

Appendix A

Scaling coefficient

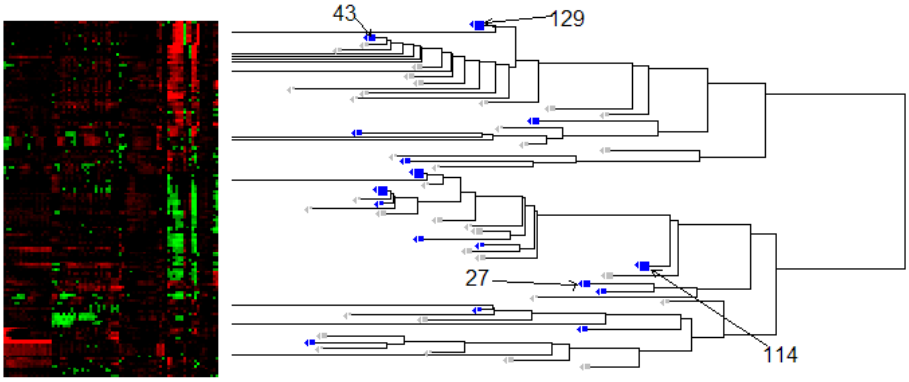


Figure A.1: The scaling of the rectangles is performed using formula $scaledSize = \lceil 1.5 * size^{\frac{1}{3}} \rceil$. As one can see, the difference between bigger (100 –200 genes) and smaller (10 – 20 genes) clusters is minimal. A lot of fine details make it hard to grasp the structure.

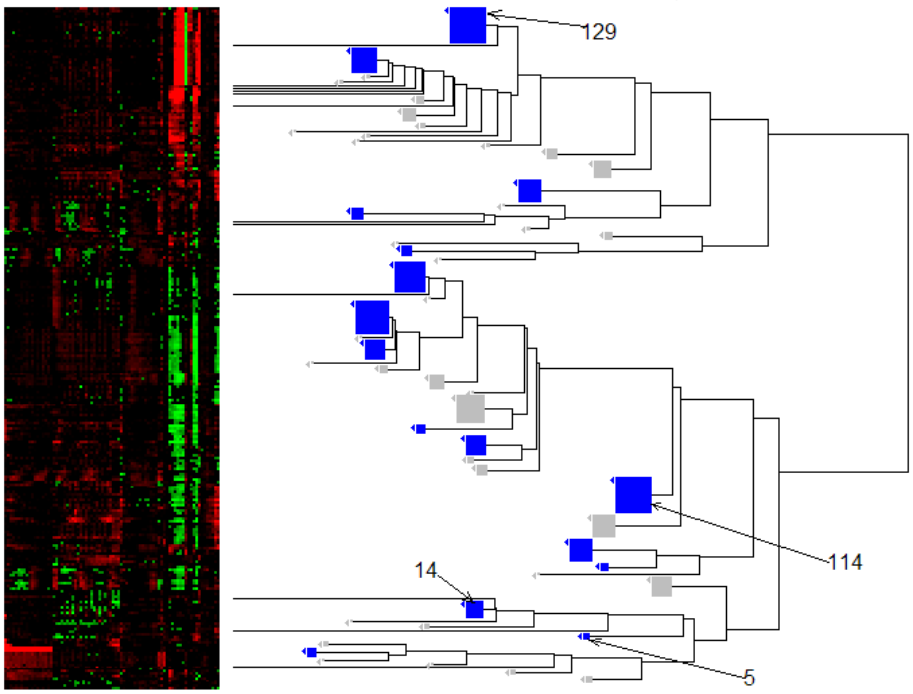


Figure A.2: The scaling of the rectangles is performed using formula $scaledSize = \lceil 5.5 * size^{\frac{1}{3}} \rceil$. Small clusters (eg 5 genes) are too small, on their background the cluster of 14 genes looks pretty big, but there is no difference between clusters that have 114 and 129 genes. The picture size for bigger data set can be too big.

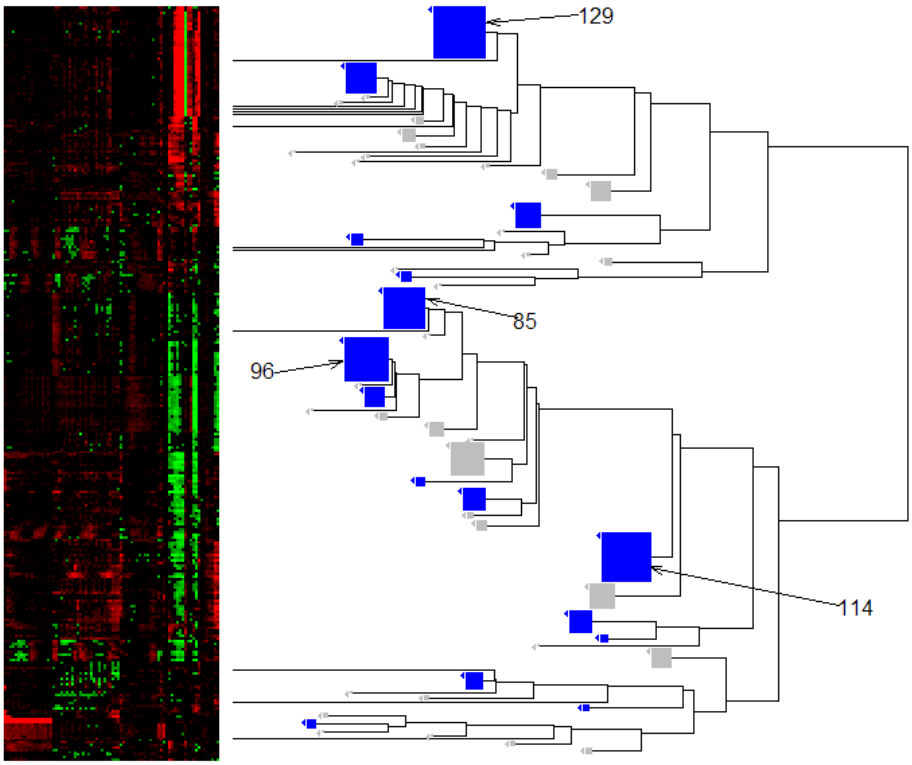


Figure A.3: The scaling of the rectangles is performed using formula $scaledSize = \lceil 3.5 * size^{\frac{1}{2}} \rceil$. Bigger clusters (129, 114, 96 and 85 genes) gain all the attention of the user. Smaller clusters cab stay unnoticed. The picture size for bigger data set can be too big.

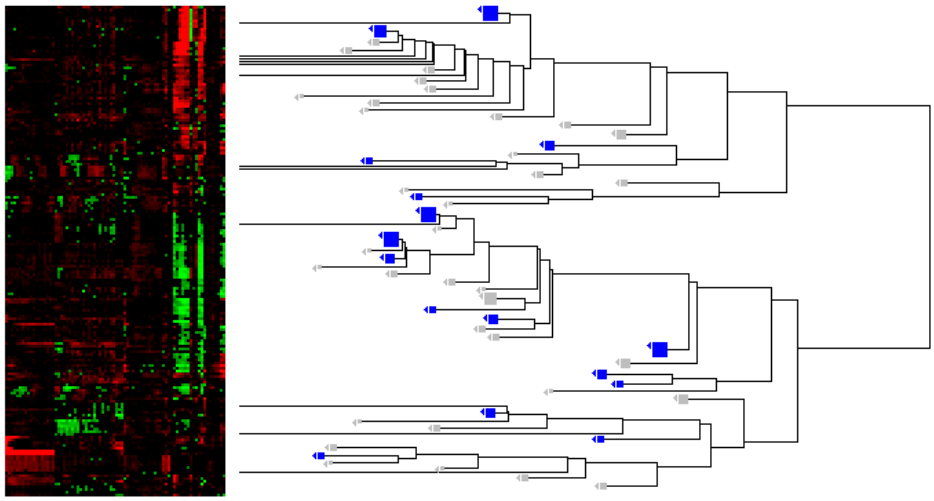


Figure A.4: The scaling of the rectangles is performed using formula $scaledSize = \lceil 3.5 * size^{\frac{1}{4}} \rceil$. All of the clusters look pretty the same, in spite of their size. A lot of fine details make it hard to grasp the structure.

Appendix B

Treeviewer GUI Screenshots

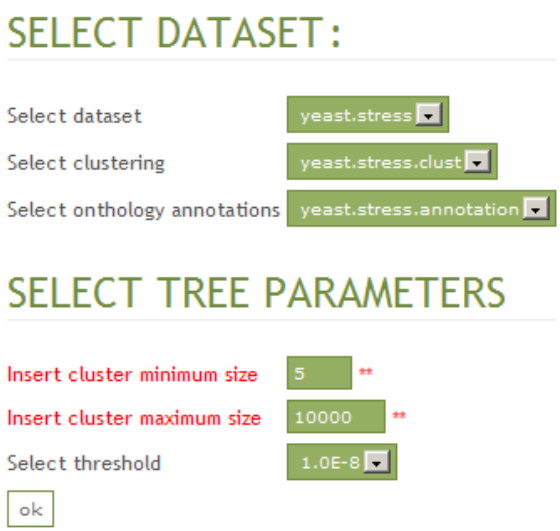
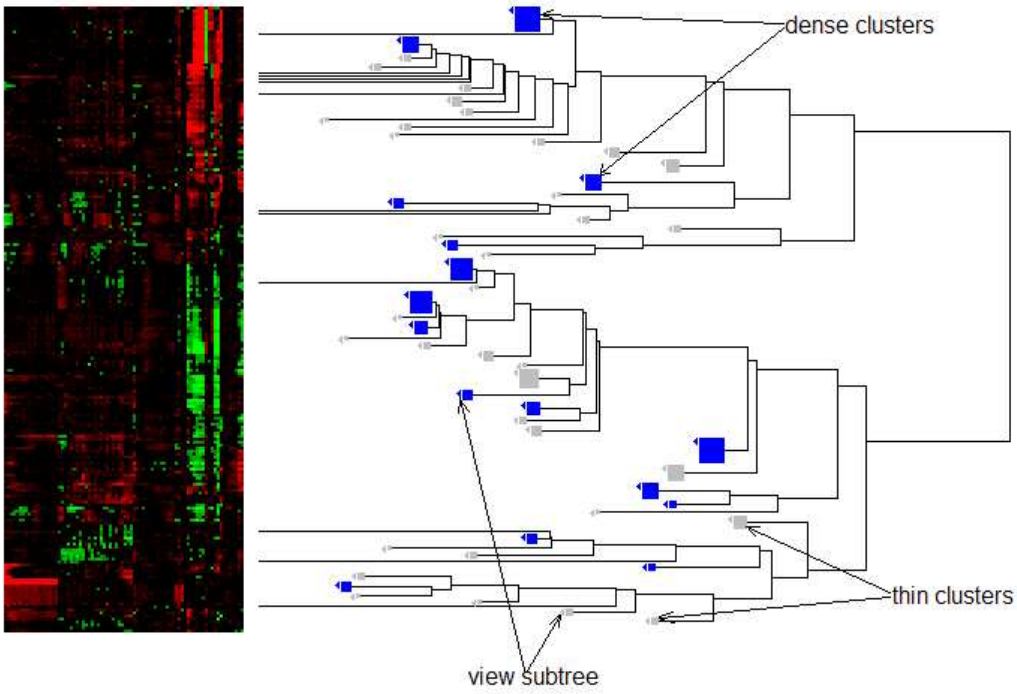


Figure B.1: Selection of data set (for the analysis and the visualisation).



list of dense clusters

CLUSTERS STATISTICS

Cluster Id	Cluster size	Best Annotation	P-value
1473	96	KEGG:03010	5.8E-126
1565	85	GO:0005730	8.01E-71
1661	129	GO:0030435	1.18E-30
1385	9	GO:0000788	3.7E-27

Serch for particular genes:

search!

List Hierarchical Clusters:

Get Hierarchical Clusters

Figure B.2: The visualisation of a single data set. The heatmap (on the left) illustrates the activities of the genes. The dendrogram represents the hierarchical structure of the data set. Blue rectangles symbolise dense and grey — thin clusters. After clicking on a rectangle of any colour, the user will be navigated to the page, that is dedicated to a single cluster. The table, containing information about dense clusters is also displayed.

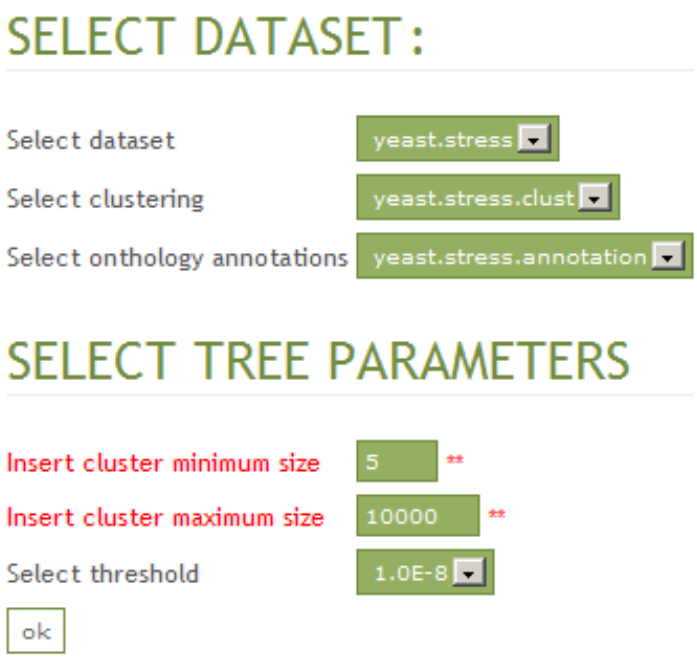
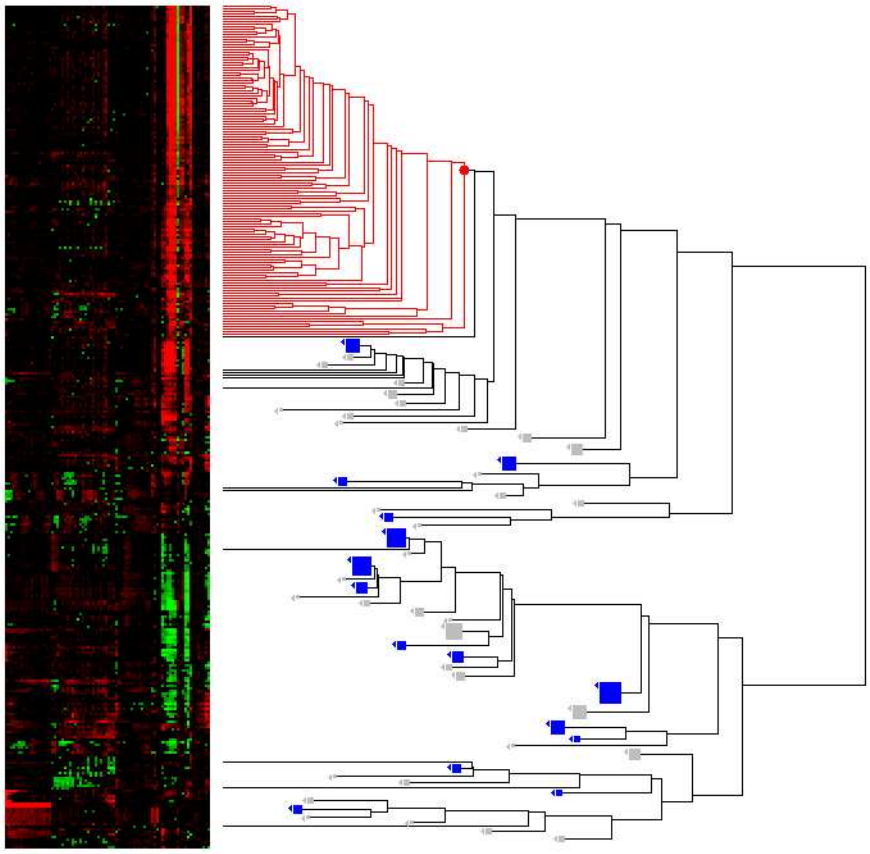


Figure B.3: Selection of data set for the single data set analysis.



CLUSTERS STATISTICS

Cluster Id	Cluster size	Best Gost	P-value
1473	96	KEGG:03010	5.8E-126
1565	85	GO:0005730	8.01E-71
1661	129	GO:0030435	1.18E-30
1385	9	GO:0000788	3.7E-27

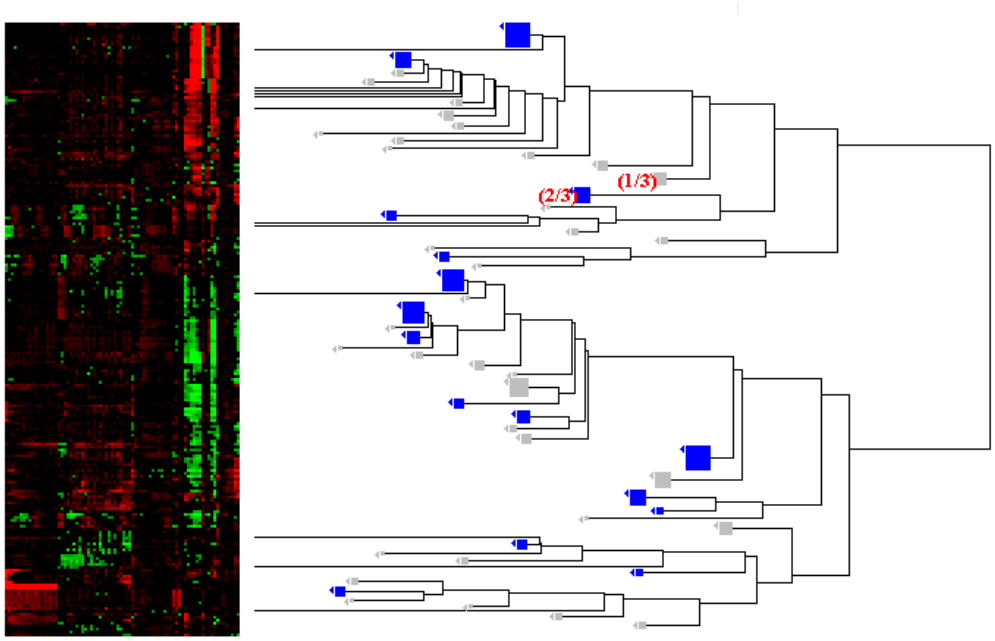
Serch for particular genes:

search!

List Hierarchical Clusters:

Get Hierarchical Clusters

Figure B.4: By clicking on the triangles user can "open" any collapsed node. To close it —- user must click on the red circle in the root of corresponding subtree. The last opened subtree is highlighted by red colour.



CLUSTERS STATISTICS

Cluster Id	Cluster size	Best Gost	P-value
1473	96	KEGG:03010	5.8E-126
1565	85	GO:0005730	8.01E-71

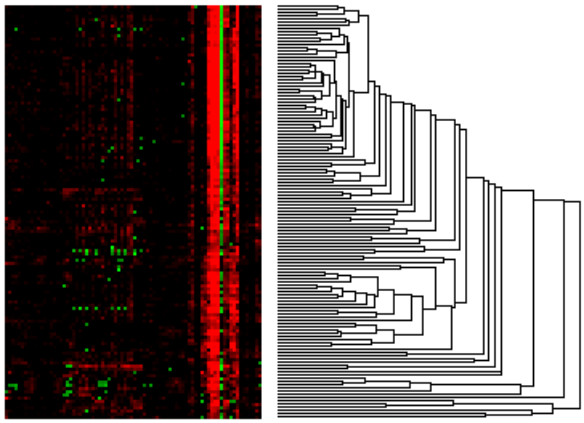
Serch for particular genes:

YFL064C YOR074C
YBR285W

search!

Figure B.5: User can search for particular gene(s) by name(s). The location of the genes will be denoted in the picture. 2/3 stands for: 2 genes out of 3 were found in this cluster.

CLUSTER ID: 1661



Cluster Information

Cluster Id: 1661

Cluster size: 129

Best Annotation: GO:0030435

P-value: 1.18E-30

g:Profiler:

Click here

gene ID converter:

Click here

orthology search:

Click here

URLMAP external pointers:

Click here

Genes in the cluster: YHR124W YOR339C YNL019C YNL033W YER115C YOR214C YGL230C YAL018C YOL047C YGL138C YDR218C YLR213C YPR027C YPL033C YBR148W YHR015W YGL170C YNL128W YFR023W YHR184W YOL132W YOR255W YDR522C YGR059W YPL130W YLR307W YLR341W YJL038C YJL037W YGR273C YOL024W YNL204C YNL318C YMR017W YOL015W YGL015C YEL023C YJL160C YOR190W YNL319W YDR523C YDR042C YOR313C YDL187C YPR078C YHR185C YFR032C YNL205C YDL114W YLR013W YER085C YDR371W YOR249C YCLX03C YBR064W YER180C YDR438W YLR343W YPL027W YDR260C YOR242C YMR125W YNL018C YIR013C YCL048W YDR402C YKL189W YPR140W YDR516C YFL011W YFL041W YML066C YDR065W YER182W YGL158W YNL270C YFL012W YDR273W YDL103C YLR049C YKL104C YFR028C YNL080C YIL159W YOR033C YNL225C YDR263C YGR225W YDR118W YJR036C YHR150W YLR030W YGR226C YGL180W YDL239C YKR005C

Figure B.6: Information about one cluster.

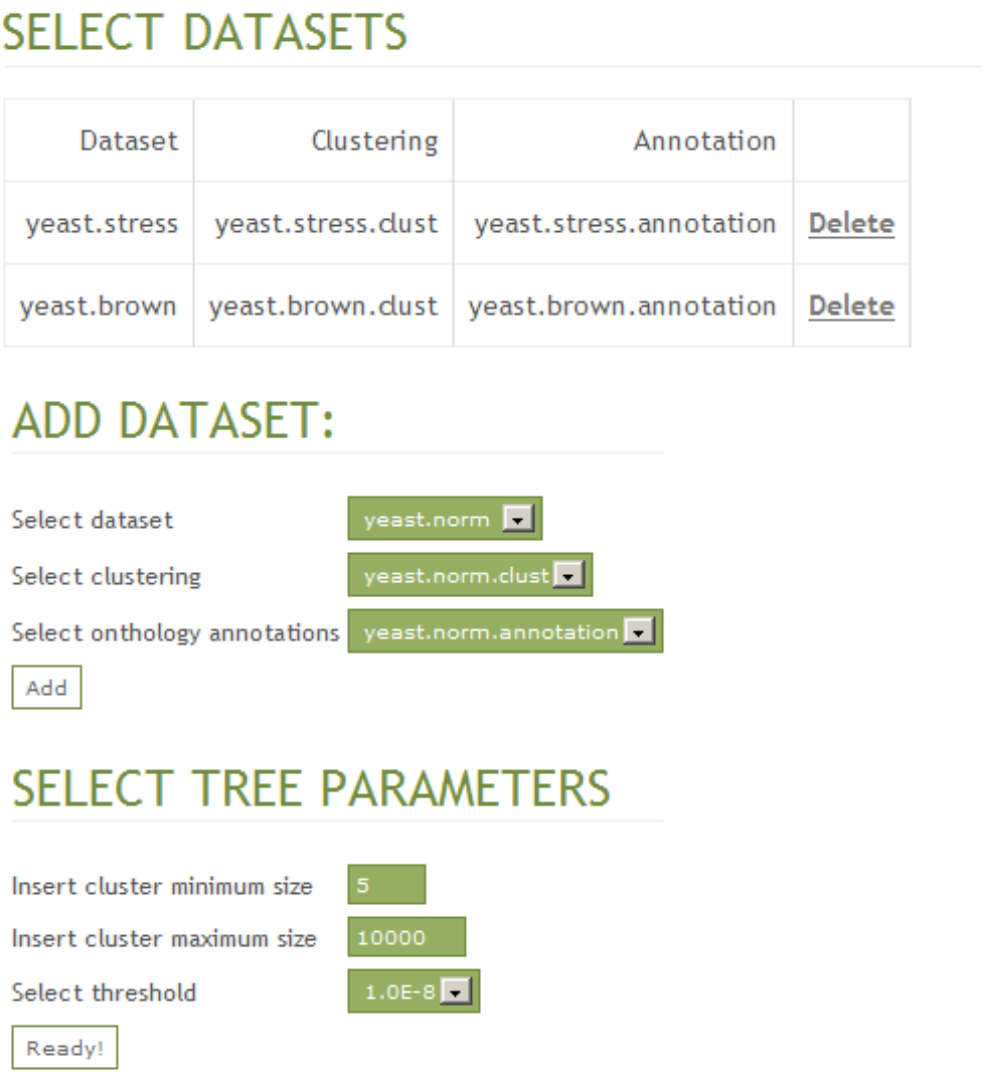


Figure B.7: Selection of two or more data sets for the comparison.

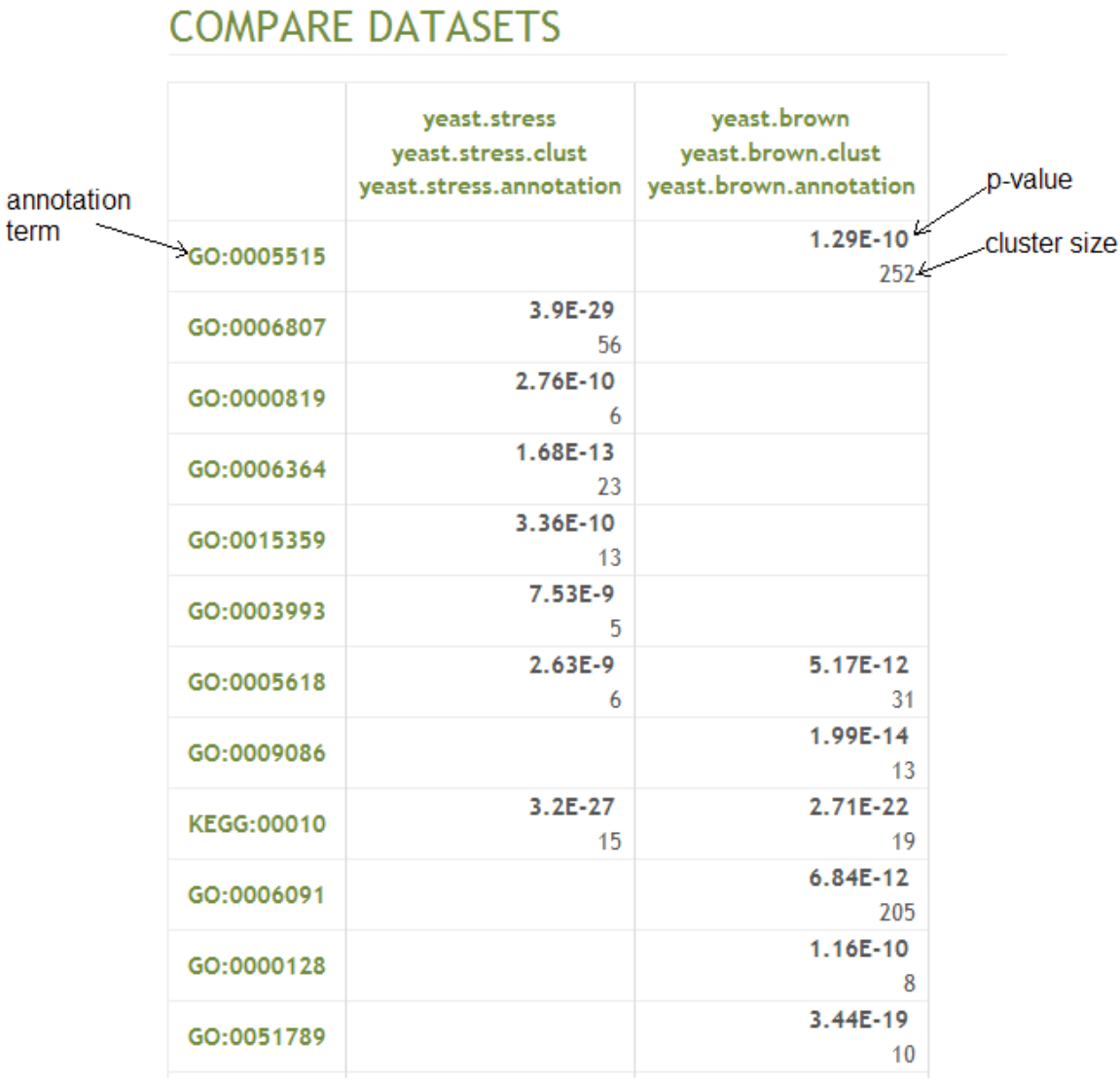


Figure B.8: The table that represents the result of comparison. The annotation terms, that were found among dense clusters of the data sets organise the rows, the data sets, that are compared, organise the columns. Each cell of the table says the p-value and size of dense cluster in particular data set and particular annotation term.