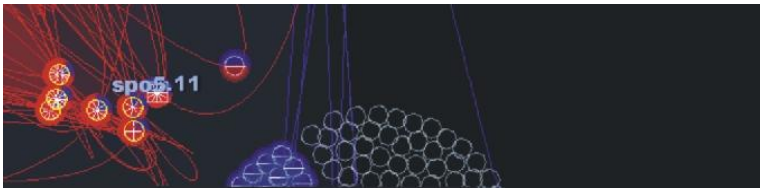
VISUAL ANALYSIS OF GENE EXPRESSION DATA BY MEANS OF BICLUSTERING

RODRIGO SANTAMARÍA VICENTE



PhD on Computers and Automation

Department of Computers and Automation
Faculty of Sciences
University of Salamanca

June 2004 - July 2009

Rodrigo Santamaría Vicente: *Visual Analysis of Gene Expression Data by means of Biclustering,* PhD on Computers and Automation, © June 2004 - July 2009

SUPERVISORS:
Luis Antonio Miguel Quintales
Roberto Therón Sánchez

LOCATION:
Salamanca

TIME FRAME:
June 2004 - July 2009

A scholar am I still to the children, and also to the thistles and red poppies.

— Friedrich Nietzsche, Thus Spoke Zarathustra, 1885

To the glowing racoons

ABSTRACT

Bioinformatics is a research area that manages large collections of data. A relevant instance of it is gene expression analysis by means of microarray technologies. The large expression matrices resulting from microarray experiments, the usually complex results of their analysis and the several sources of related existing knowledge that can be used for support and validation, demand to use the full cognitive capabilities of the analyst, not only abstract but also perceptual. Non-supervised data mining techniques contribute to discover new information, frequently in the form of groups or classifications to be inspected. Specially, in the case of biclustering methods, there are no satisfactory visualization techniques to inspect these groups (biclusters). This thesis approaches the mentioned issues by developing novel visualization techniques and integrating several sources of data, biclustering algorithms and visualizations into a single analysis framework. This framework will contribute to make it easier and improve the reasoning process related to gene expression analysis by means of a visual analytics approach.

RESUMEN

La bioinformática es un área de investigación que maneja grandes colecciones de datos. Un ejemplo importante es el análisis de la expresión genética mediante microarrays, donde están involucradas no sólo las grandes matrices de expresión derivadas de experimentos con microarrays, sino también los (a menudo) complejos resultados de su análisis y la cantidad de fuentes externas de conocimiento que se pueden utilizar para confirmar o validar los resultados del experimento. Esta cantidad de información hace necesario el uso de todas las habilidades cognitivas del analista, no sólo abstractas, sino también perceptivas. El uso de técnicas de minería de datos no supervisadas contribuye a descubrir nueva información a partir de los datos de expresión, normalmente en forma de grupos o clasificaciones. Especialmente, los métodos de biclustering se han utilizado con bastante éxito para este fin. Sin embargo, no existen técnicas de visualización satisfactorias para inspeccionar los resultados del biclustering ni para integrarlos con el resto de información disponible. Esta tesis se acerca a estas cuestiones desde el punto de vista de la analítica visual, buscando nuevas técnicas de visualización para representar biclusters e integrarlos, junto con las matrices de expresión y las fuentes externas de conocimiento, en un entorno de trabajo que facilite el proceso de análisis.

PUBLICATIONS

Some ideas and figures presented in this thesis have appeared previously in the following publications:

Santamaría, R.; Therón, R. and Quintales, L. (2008), *A visual analytics approach for understanding biclustering results from microarray data*, BMC Bioinformatics 9(247).

Santamaría, R.; Therón, R. and Quintales, L. (2008), *BicOverlapper: A tool for bicluster visualization*, Bioinformatics 24(9), 1212–1213.

Chen, Y.; Santamaría, R.; Butz, A. and Therón, R. (2009), *TagClusters: Semantic Aggregation of Collaborative Tags beyond TagClouds* in Lecture Notes in Computer Science. Smart Graphics 2009, Springer Verlag, pp. 56–67.

Santamaría, R. and Therón, R. (2008), *Overlapping Clustered Graphs: Co-authorsip Networks Visualization*, in Lecture Notes in Computer Science. Smart Graphics 2008, Springer Verlag, pp. 190–199.

Santamaría, R.; Quintales, L. and Therón, R. (2007), *Methods to bicluster validation and comparison in microarray data*, in Lecture Notes in Computer Science. IDEAL 2007, pp. 780–789.

Santamaría, R.; Therón, R. and Quintales, L. (2007), *A Framework to Analyze Biclustering Results on Microarray Experiments*, in Lecture Notes in Computer Science. IDEAL 2007, pp. 770–779.

ACKNOWLEDGMENTS

I wish to thank my brother and my parents for being always there, helping without conditions and keeping my feet on earth.

Many thanks to Luis, because of his patient dedication and his invaluable guidance on the road of bioinformatics. Also many thanks to Roberto, for helping to get the best of me and for his invaluable guidance on the road of information visualization.

Also I wish to say thank you to the members of the VisUsal group: Juan, Diego, Vadim, Carlos and Antonio. We started together our PhDs and collaborated in coding some visualization techniques for the InfoVis'07 and GraphDrawing'07 contests. I don't want to forget to thank Javier Molpeceres and Antonio Hernández, authors of the projects that served as starting points for BicOverlapper and Treevolution, respectively.

Thanks to all the people at the University of Salamanca, specially at the Computers and Automation Department, that make possible this thesis. I wish to thank in a special way to Iván, because of his help with the lectures, and my 'flatmates' Carlos, Susana and Juan Carlos. And many thanks too to Francisco Antequera from the IMB for his biological advice and help. I also want to say thanks to Begoña for giving me the opportunity to start a scientific career, and also to the rest of the LRI and the Nuclear Physics Group, specially to Francisco Fernández, Alfredo, Felipe, María, Jorge, José Ramón, Fuensanta and Paula.

Many thanks to Misha Kapushesky and Alvis Brazma for giving me the fantastic experience of working at the EBI. Also thanks to Helen Parkinson, Tony Burdett, Juok Cho, Audrey Kauffmann, Ekaterina Pilicheva, Anjan Sharma and Nils Gehlenborg for their collaboration at the EBI and their feedback about the bioinformatics of my thesis. Also thanks to Ibrahim, Roby, Eamonn, Mike, Nikolay, Pavel and so many other great people there in Cambridge.

Thank you to Sebastian Kaiser and Friedrich Leisz from the University of Munich for letting me to collaborate in their R project for biclustering algorithms.

Last but not least, thanks to my friends, Alfonso, Carlos, María, Celia, Seve, Ana and a long etcetera of good friends which supported me and with which I enjoyed so many good moments.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

A Adenine

ACM Association of Computing Machinery

ADF Array Design Format

AE ArrayExpress

ANN	Artificial Neural Networks
API	Application Programming Interface
BicAT	Biclustering Analysis Toolbox
BP	Biological Process
C	Cytosine
CC	Cellular Component
CandC	Cheng and Church biclustering algorithm
cDNA	complementary DNA
CESR	Core Environmental Stress Response
CGH	Comparative Genomic Hybridization
ChIP-on-chip	Chromatin InmunoPrecipitation on gene chip
CIBEX	Center for Information Biology gene EXpression database
ChroCoLoc	Chromosome Co-Localization
CLICK	CLustering Identification via Connectivity Kernels
C/T	Cretaceous-Tertiary
DLDA	Diagonal Linear Discriminant Analysis
DNA	DeoxyriboNucleic Acid
EBI	European Bioinformatics Institute
EF	Experimental Factor
EFO	Experimental Factor Ontology
EFV	Experimental Factor Value
Expander	EXPresion ANalyzer and DisplayER
FDCG	Force-Directed Clustered Graph
FLOC	FLexible Overlapped biClustering
G	Guanine
GEMS	Gene Expression Mining Server
GEO	Gene Expression Omnibus
GO	Gene Ontology
GOA	Gene Ontology Annotation
HCE	Hierarchical Clustering Explorer
HCG	Hierarchical Clustered Graph
HCI	Human-Computer Interaction
HCIL	Human-Computer Interaction Lab

HM HeatMap

IDF Investigation Description Format

IEEE Institute of Electrical and Electronics Engineers

IMB Instituto de Microbiología Bioquímica

ISA Iterative Signature Algorithm

InfoVis Information Visualization

LRI Laboratorio de Radiaciones Ionizantes

KDD Knowledge Discovery and Data mining

KEGG Kyoto Encyclopedia of Genes and Genomes

KNN K-Nearest Neighbors

LD Linkage Disequilibrium

LDA Linear Discriminant Analysis

LOWESS LOcal Weighted regrESSion

MAGE MicroArray Gene Expression

MDS MultiDimensional Scaling

MGED Microarray Gene Expression Data

MIAME Minimal Information About Microarray Experiments

MMS Methylmethane sulfonate

MRI Magnetic Resonance Imaging

NCBI National Center for Biotechnology Information

NVAC National Visualization and Analytics Center

OPSM Order-Preserving SubMatrix

PC Parallel Coordinates

PCCC Pearson's Cophenetic Correlation Coefficient

PCA Principal Component Analysis

PCR Polymerase Chain Reaction

PPI Protein-Protein Interaction

PTM PostTranslational Modification

QDA Quadratic Discriminant Analysis

RGD Rat Genome Database

RNA RiboNucleic Acid

RPM Rich Probabilistic Model

MF Molecular Function

mRNA	messenger RNA
SAMBA	Statistic-Algorithmic Method for Bicluster Analysis
SDRF	Sample and Data Relationship Format
SESR	Specific Environmental Stress Response
SGD	Saccharomyces Genome Database
SNP	Single Nucleotide Polymorphism
SOAP	Simple Object Access Protocol
SOM	Self-Organizing Maps
SVD	Singular Value Decomposition
SVM	Support Vector Machines
T	Thymine
TRN	Transcription Regulatory Network
U	Uracil
VAST	Visual Analytics Science and Technology
WC	Word Cloud

Part I

INTRODUCTION

1 INTRODUCTION

It is not enough to have a good mind. The main thing is to use it well. — René Descartes, Discourse on Method, 1637

The past decade has witnessed the advent of a lot of achievements within the field of genomics. Initiatives such as the Human Genome Project [138, 69, 1] and similar projects for other organisms [18, 2, 143] have established the basis for the genetic structure of several key organisms by identifying DNA sequences as genes. Although far from perfection by themselves, these sequence-to-gene mappings are enough to dramatically increase our understanding of genomics.

the best estimates indicate that about 92% of the human genome has been completed, and characteristics such as junk DNA are still not understood

The combination of gene-to-sequence mappings and gene manipulation technologies developed after Polymerase Chain Reaction (PCR)[1], lead to multiple technologies to record the behavior of genes under different conditions. The most used ones are *microarray technologies*, which manage to measure the amount of transcription for several gene sequences. Each microarray typically handles every known gene of an organism (which means thousands of genes), and usually several microarray experiments are conducted (regarding different conditions or just replications), so it is normal to work with millions of transcription values.

The technology to build microarrays has evolved so much in the past five years that today all the genome sequence of an organism can be included in a single microarray. This evolution of technologies leads to an increase in the amount of data to analyze. In addition, the spread and commercialization of microarray technologies, and the use of public repositories increase the number of conditions under which a given microarray platform is utilized, therefore enlarging the amount of data to analyze.

microarrays comprising whole genomes are called tiling arrays, *each* tile *is a sequence of around 25 nucleotides a single tiling array can have millions of probes*

A fact that is shared by almost any research field, not only genomics, is that there is a *bottleneck* affecting our capability to analyze such vast amounts of data provided by existing technologies. Fortunately, there is a large number of analysis techniques available to the researcher in order to filter, group and classify information. The use of some of these analysis techniques is complex and frequently simple techniques prevail. Simpler techniques usually disregard several details on the data, but there is so much information on these datasets that it can be enough to find important discoveries. On the other case, specially in exploratory analyses, the scientists demand more complex methods, but usually these methods give us too much information and their results are hard to inspect. This is the case, regarding gene expression analysis, of *biclustering algorithms*; an evolution of traditional clustering techniques that has become popular because their design fits better to biological behavior.

In the case of the analysis of microarray data, in addition to the inspection and interpretation of the outcomes of analysis techniques, there is available information about genes and gene transcription that is

1 PCR is a technique that uses DNA polymerase to make multiple copies of a piece of DNA

used to validate the analysis and interpret microarray experiments. For example, Transcription Regulatory Networks (TRN) convey transcription regulation among genes. Gene annotations store biological knowledge related to genes, from its chromosome to the molecular functions into which they are known to be involved. Biological pathways relate genes that work together in a given biological process. Bioinformaticians and biologists make use of several tools and web services that provide such kind of information.

A broadly accepted approach to this problem of excess of information is to widen the intelligence levels used, adding visual thinking to abstract cognition [141]. This approach gave way in the past decade to Information Visualization, that has revealed as a key research area to guide and increase the capabilities of different analysis techniques by means of visual representation and interaction. It has, for example, become a standard *de facto* to visualize gene clusters on a heatmap [44]. In order to cover the whole analytical process, not only the visual display of data, Visual Analytics [130] is spreading among different application areas, and has emerged as a research area by itself.

1.1 MOTIVATION

The design and analysis of microarrays is directed to answer several questions, that can be summarized as [20]:

- How does gene expression level differ in various cell types and states, how is gene expression changed by various diseases and compound treatments?
- How are genes regulated, how do genes and gene products interact, what are these interaction networks?
- What are the functional roles of different genes and in what cellular processes do they participate?

The analytic discourse to answer these questions depends on the availability of previous knowledge and the extent of the questions. Instances of these questions such as *do the transcript abundances of cancer related genes for a given patient match the normal abundances?*, answered with yes/no, are frequently useful in biomedical applications. This kind of reasoning comprised of hypothesis testing usually requires a low number of conditions (disease/control) and relatively simple analysis techniques (such as differential analysis [6]).

Broader questions such as *which genes are involved in the cellular response to stress?* fit better with an exploratory analysis, which requires larger experiments (several conditions with different kinds of stress) and more complex analyses (non-supervised classification).

In the case of hypothesis testing, the available biological knowledge directs the discourse, while in exploratory analysis it is just a guide to support or validate discoveries, that will generate new biological knowledge. Although hypothesis testing has really important applications such as diagnosis, the present work is more dedicated to exploratory analysis, which requires of more complex analysis techniques.

However, as of today, some of these complex analysis techniques, specially biclustering, are not widely used in practice, the more traditional

clustering being preferred, despite the consensus in the theoretical advantage of the former [84, 126, 99].

It is our aim to analyze the reasons for this gap between theory and practice and to make biclustering a more utilized option in exploratory analysis. As an advance, some of the issues related with biclustering analysis are:

- Biclustering algorithms are heterogeneous about the kind of groups they search for and the search methods.
- There are no golden rules to compare biclustering goodness.
- The biological interpretation of some kinds of biclusters is unclear.
- There are few comprehensive compilations of biclustering algorithm implementations.
- There are few specific visualization techniques and tools for biclustering.

1.2 AIM OF THE THESIS

The first part of this work focuses on the last two issues enumerated above: to compile biclustering algorithms and study bicluster visualization; which eventually leads to provide arguments to face and discuss about the first ones. The reason to proceed in this way is that the variety of methods and definitions of what is a bicluster makes very difficult to design a golden numerical metric to validate them all. Regarding this issue, this work proposes a metric applicable to the computation of the best parameter setting for a biclustering algorithm, a first step towards biclustering benchmarking. This work also proposes a novel visualization technique to represent biclustering results making emphasis in conveying the special properties of biclusters.

On the second part of the work, our aim is to step back and watch to the whole reasoning process associated to gene expression analysis, integrating biclustering algorithms, visualization techniques and external knowledge on a framework for visual analysis. The result is a tool that helps the analysts to explore their data, reducing time and effort and boosting their analytical capabilities thanks to *ad hoc* visual representations and a high interaction with the tool. This tool has been tested with several biological examples to perform gene expression analysis based on biclustering in order to discover biological knowledge and to compare it with other analysis approaches.

1.3 CONTRIBUTION TO KNOWLEDGE

The main outcomes of the work described in this thesis are:

- BicOverlapper [106, 107], a framework that consistently applies a visual analytics approach to the exploration of gene expression data with biclustering algorithms. BicOverlapper is available at: `http://vis.usal.es/bicoverlapper`
- A general study and review of visualization techniques in the field of gene expression analysis and biclustering, with the description of our proposed improvements and a novel bicluster visualization technique.

There is a number of minor contributions derived from the iterative research cycle. These are:

- The development of metrics for the internal validation of biclustering algorithms and the tuning of its configurable parameters.
- The development of *biclust*, a R package with the implementation of several biclustering algorithms. This has been done in collaboration with Sebastian Kaiser and Friedrich Leisch from the University of Munich, and is available at: `http://cran.r-project.org/web/packages/biclust`
- The description of a formal approach to the design of solutions for the visual analysis of gene expression.
- The discussion of several case studies of gene expression analyses by means of biclustering.

Hopefully, these contributions will lead to the improvement and spreading of biclustering analysis and gene expression analysis in general, and to a more frequent usage of visual analytics approaches in biological research.

1.4 ORGANIZATION OF THE THESIS

The *Background Information* chapter summarizes the main relevant concepts regarding to the four major research fields involved in this work: microarray bioinformatics, gene expression data analysis, information visualization and visual analysis. The *State of the Art* chapter describes the specific techniques related to this thesis: biclustering analysis, visualization of microarray data and biclustering results, and visual analysis on bioinformatics. The *Problem Statement* chapter illustrates the issues regarding to biclustering analysis on gene expression data. This work focuses on the validation and parameter configuration of biclustering algorithms and, specially, in the visualization of biclustering results and the application of a visual analytics approach to the study of gene expression. The *Proposed Solution: Design* chapter describes in detail how we approached the above problems and developed solutions for them. The *Proposed Solution: Results* chapter presents the use of the designed solutions in practical cases in order to confirm their usefulness. Finally, the *Conclusions* chapter discusses the achievements of this work, the new problems that arise and future work lines in order to solve them.

Part II

BACKGROUND INFORMATION

2 MICROARRAY BIOINFORMATICS

Science consistently produces a new crop of miraculous truths and dazzling devices every year. — Kary Mullis

Bioinformatics is the application of information technology to molecular biology (the study of biology at a molecular level). The term was coined in 1978 by Paulien Hogeweg and has its relevance has increased since then. Due to the high amounts of information related to molecular biology technologies, bioinformatics has become essential in certain areas, such as management of DNA and protein sequences, gene expression analysis, protein structure analysis, etc.

The actual process of analyzing and interpreting biological data is referred to as *computational biology*, so bioinformatics is, from this point of view, part of computational biology[1]. Anyway, bioinformatics and computational biology coincide in several sub-disciplines, such as the use of tools that enable efficient access and management of biological information and the development of algorithms to analyze and search for relationships within these data.

In this chapter, we will review some concepts regarding microarray bioinformatics that are key to identify the design requirements of gene expression analysis, either visual or algorithmic. Likewise, these concepts will help to understand the nature of input data and to evaluate the results of the analysis process.

2.1 GENE EXPRESSION

Along this and the following chapters, a number of biological terms will appear related to gene expression. It is vital, in order to design a good approach to gene expression analysis, to understand, at least at a basic level, its biological grounds.

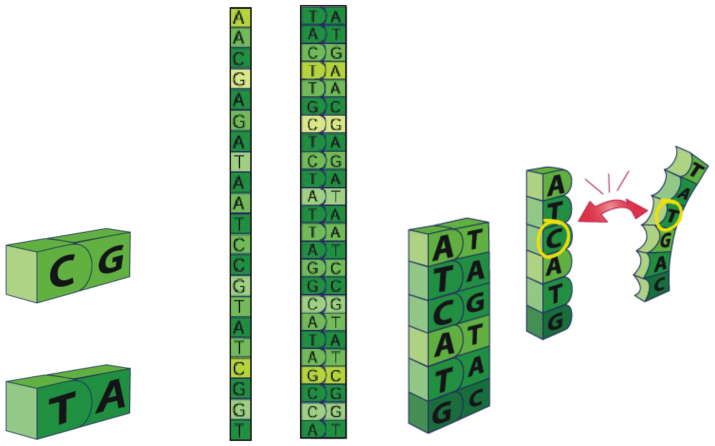
2.1.1 *Nucleotides, Genes and Proteins*

Any information about a living organism is coded in complex combinations of four structural units called *nucleotides* or bases: Adenine (A), Guanine (G), Cytosine (C) and Thymine (T). Nucleotides are molecules with the characteristic that each one is chemically attracted by another nucleotide and repelled by the other two, forming the base pairs AT and CG. This attraction/repulsion property is applicable to large nucleotide chains, not only single base pairs (see fig. 1).

Thymine is substituted for Uracil (U) in RNA

DeoxyriboNucleic Acid (DNA) and RiboNucleic Acid (RNA) are molecules that consist on a long nucleotide chain. RNA is usually single stranded, while DNA is usually double stranded (see fig. 1b). DNA serves as the storage for all information (genes or not) of an organism, while RNA, among other functions, acts as a bridge to transform genes into proteins or other gene products.

1 This is analogous to the relationship between information visualization and visual analytics, that will be discussed in 5.



(a) (b) (c)

Figure 1: a) Base pairs. b) Single and double-stranded nucleotide sequences. c) Illustration of how two complementary sequences are attracted and how two sequences with just an inadequate pair combination repel each other (reproduced from Affymetrix web page).

In general terms, a *gene* is a nucleotide sequence that codifies a trait of a living organism. For example, there is a gene for the eye color, so an instance of the gene present on a specific individual (called *allele*) determines the eye color for that individual. The allele set of an individual is called the *genotype* and the set of traits it codifies is the *phenotype*. The gene set of a species is called its *genome*, and depending on the kind of organism, it will be organized into *chromosomes* (*eukaryote*) or not (*prokaryote*).

chromosome means "colored body" in Greek, because they are strongly stained by some dyes

A *protein* or polypeptide is a long (50 to 1000) chain of amino acids. Amino acids are the structural molecular units of proteins, analogously to nucleotides in genes. There are 20 different amino acids, each one is the result of the combination of three nucleotides[2].

2.1.2 *Transcription, Translation and Expression*

Basically, *transcription* is the copy of a nucleotide chain (DNA) to another (RNA) (see fig. 2). It is based on *hybridization*, a process by which two complementary single chains become a double chain due to the base pair affinities. *Translation* is the conversion of RNA chains into amino acids. *Gene expression* is the process by which a gene is made into a functional gene product, usually a protein. The whole process is known as the *central dogma of molecular biology*:

Ribosomes are responsible of translation

Proteins are translated from RNA, which is transcribed from DNA

Microarray data analysis is based on the fact that, if there are lots of copies of a gene due to transcription, there will be lots of RNA chains to translate it into amino acids and therefore lots of proteins to express the corresponding gene trait.

The regulation of gene expression (or just *gene regulation*) includes any process that cells use to turn the information of genes into gene

2 There are 64 possible combinations of nucleotides, the existence of just 20 amino acids means a lot of redundancy

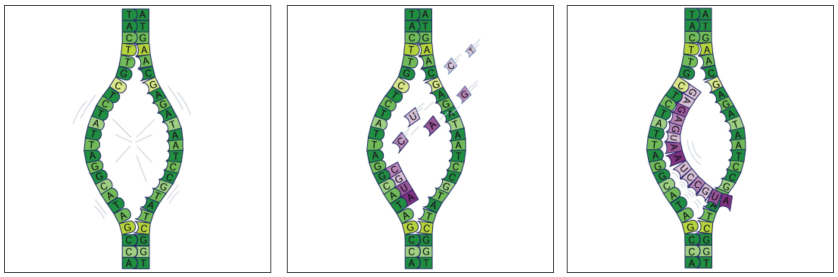


Figure 2: A simplified process of DNA transcription. RNA polymerase separates a portion of DNA (left), then attaches nucleotides thanks to their chemical attraction (center) to finally obtain the desired mRNA (right) (reproduced from Affymetrix web page).

products. It includes translation and transcription, but it is important to note that there are some other processes that modify gene expression, such as RNA transport, messenger RNA (mRNA) degradation or PostTranslational Modification (PTM), although they are out of the scope of this introductory chapter.

2.2 MICROARRAYS

Microarray is a term referring both to a technology and the result of its application (the *microarray chip* or gene chip). It consists in arrayed series of thousands of microscopic spots filled with nucleotide sequences that measure their transcript abundance.

Microarray *platform* or microarray *design* refer to the architecture for a specific organism or purpose. For example, the platform *hgu95av2* is one of the architectures of Affymetrix company for *Homo sapiens*. Microarray *experiment* refers to each biological experiment applied under a given platform, usually involving several microarray chips of a given platform. Microarray *data* are the results of such an experiment, frequently resumed in a *gene expression matrix*. Sometimes all these terms (technology, chip, platform, data) are described just as *microarray*, leaving context to clarify any possible ambiguity.

This chapter briefly overviews the most important aspects of microarrays, please refer to [16, 61] for detailed coverage of the matter.

2.2.1 *Microarray Technology*

gene chips are usually made of glass or silicon

A microarray consists of a solid surface, known as *gene chip*, where genetic material is placed. The gene chip has a grid-like structure, each *spot* containing a different single strand nucleotide sequence known as *probe*. Each spot contains millions of copies of its probe. There are several kinds of microarray. Following, we describe the microarray building process for a complementary DNA (cDNA), one-channel chip (see section 2.2.3 for more information about the kinds of microarrays).

1. A sample coming from an organism under the experimental condition to test is prepared. It contains cDNA sequences for each probe in the gene chip, but with a *fluorescent* label added to each sequence[3].

3 Note that in the sample the number of copies of each sequence is directly related to the transcript abundance of that sequence, while the number of copies of each sequence in the gene chip is the same for all the probes

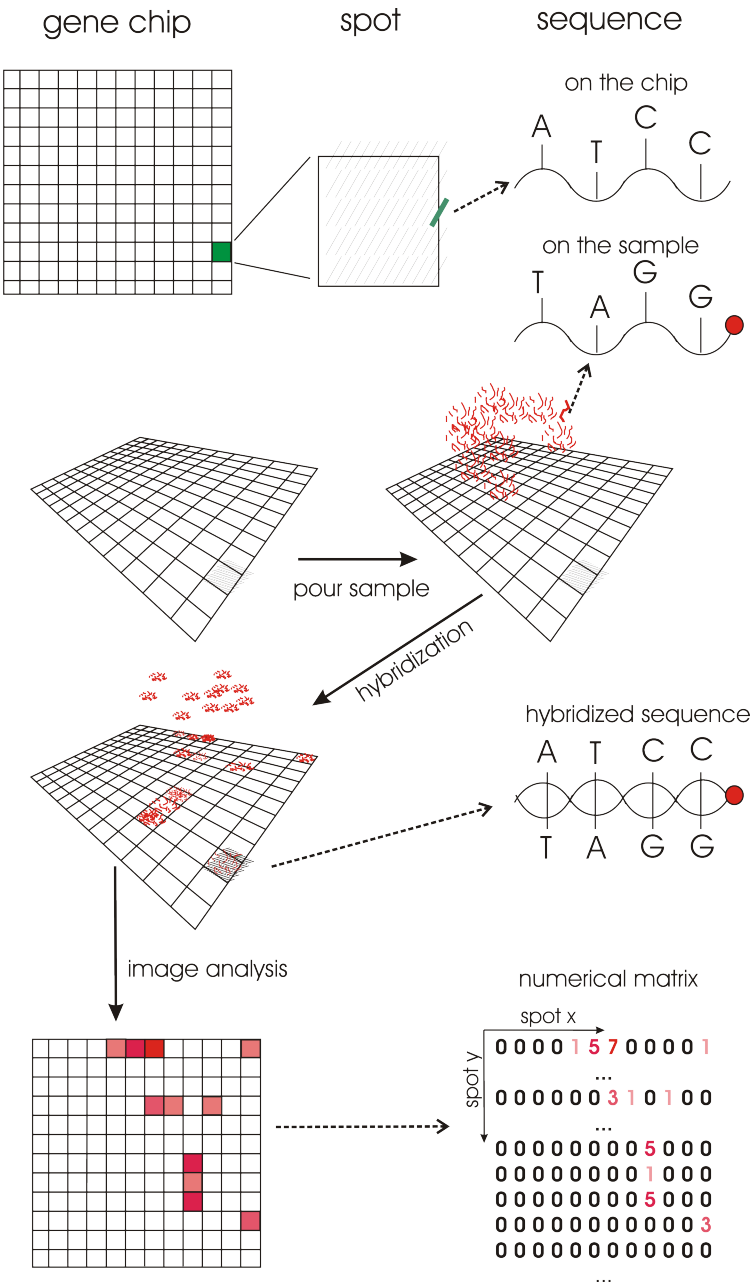


Figure 3: Microarray build and image analysis processes.

2. Afterwards, the sample solution is poured onto the gene chip and, by means of hybridization, the matching sample sequences will attach to their complementary chip sequences.

3. The gene chip is then washed and dried, so finally each spot will contain a different number of "stuck" sample sequences, and therefore a different number of fluorescent labels.

4. The chip is now read by stimulating the fluorescent labels and measuring the light intensity of each spot, being this intensity proportional to the transcript abundance of the sample. This image analysis is not trivial and requires procedures such as pixel detection, background intensity correction, intensity bias correction, etc.

Affymetrix provides intensity matrices in CEL files, and the chip design in CDF files that provide the probe-to-gene-mappings (at least 11 probes per gene)

The whole process of microarray building is illustrated in the schema of fig. 3. After the image analysis is performed, we have an *intensity matrix* I, being I_{ij} the light intensity of the spot located at the position (i, j) in the array, which corresponds to the light intensity of a probe sequence. Usually, there are more than a single probe related to a gene, so in order to obtain the gene levels from the probe levels, we need to perform a summarization of probe levels.

2.2.2 *Microarray Experimental Designs*

Microarray experiments usually comprise two or more samples. A typical paradigm is *control vs. disease*, where two samples are prepared, one with a healthy or otherwise normal individual (*control*) and another one with an individual under a disease or other relevant condition (*disease*). The result is an experiment requiring two microarrays or, in other words, two intensity matrices. If the same chip design is used for both experiments, the intensity matrices can be converted into arrays and stuck together in a *gene expression matrix* A, with n rows (one per gene) and two columns (control and disease). Element a_{ij} is the expression level of gene i under condition j. The conditions are often replicated to check experimental variance, resulting into four or more columns on the gene expression matrix. In addition, larger experiments may include several different conditions, each one with replications, increasing the number of columns of the matrix. Fig. 4 shows how various probe intensity matrices produce the final gene expression matrix.

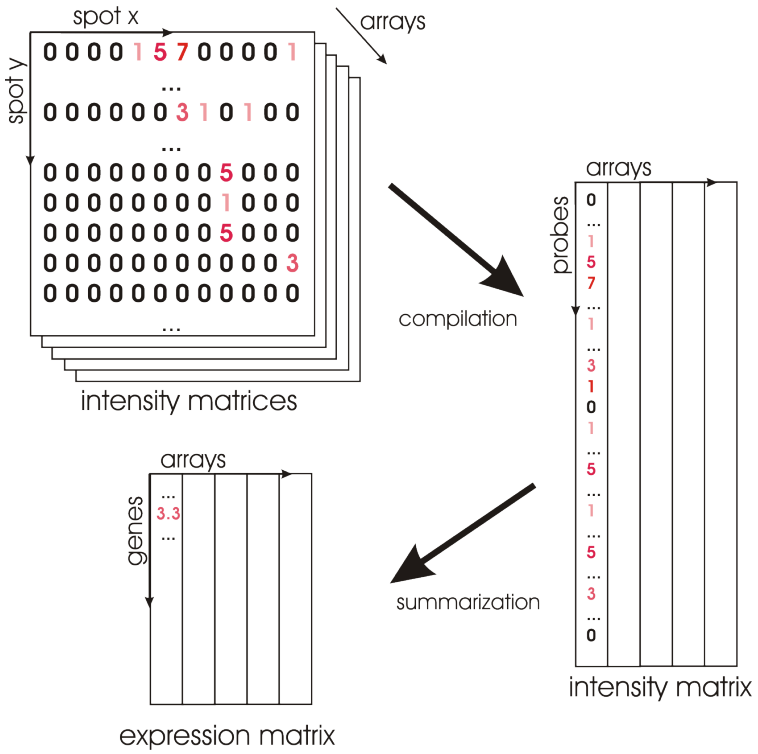


Figure 4: Microarray compilation and summarization. The numbers in red correspond to several probes for a single gene.

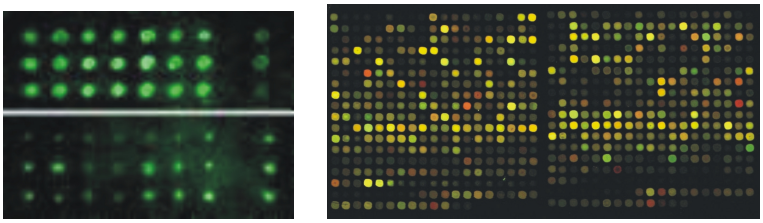
Although unusual, some experiments may use more than one array design, called multiarray design *experiments*

Microarray experiment refers to the whole experimental design. The words *condition*, sample, array or assay refer to the samples (each of the columns in the gene expression matrix). *Array design* or microarray platform refer to the specific chip used in the microarray experiment. The data matrix of a microarray experiment is usually named *gene expression matrix* or just expression matrix.

2.2.3 *Types of Microarrays*

The philosophy of microarrays is almost the same for every microarray, but slight modifications in the above procedure lead to different types of microarrays. Regarding the number of samples poured over the gene chip we have (see fig. 5):

- *One-channel microarray*: genetic sequences from one sample are poured over the microarray for hybridization. An example of one-channel microarrays are Affymetrix GeneChips.
- *Two-channel microarray*: genetic sequences from two different samples are poured over the microarray for *competitive* hybridization. Each one is marked with a fluorescent label of a different color. Two-channel microarrays are also called Cy3/Cy5 microarrays, because these two kinds of cyanines are used as fluorescent dyes for the channels.



(a) One-channel microarrays (b) Two-channel microarrays

Figure 5: One and two-channel microarrays after fluorescent stimulation.

Two-channel technology requires only one chip in order to perform a simple control vs. disease experiment, while one-channel technology needs two, and the comparison is made computationally instead of experimentally. However, the power of one-channel microarrays is that, if done under the same instruments, protocols and design (which is common with company-specific chips), every experiment performed on the same chip series are comparable among them, adding tremendous power to the experimental design.

Regarding probe sequence origin, we can distinguish:

- *cDNA microarray*: probes are cDNA sequences.
- *oligonucleotide microarray*: probes are short nucleotide sequences specifically synthesized for the experiment.

Finally, depending on probe preparation, we have:

- *delivery microarray*: probes are prepared off-line using techniques such as cloning or Polymerase Chain Reaction (PCR) and then are delivered onto the gene chip by contact printing. They are often called spotted microarrays.

- *in situ microarray*: probes are prepared directly onto the gene chip, adding nucleotide by nucleotide until the desired sequence is complete, by means of photolithography.

Delivery technology is cheaper and can produce microarrays of medium density[4]. In situ technology is more expensive but can produce microarrays of higher density, such as *tiling arrays*. The probes in a tiling array cover the *whole* genome of the organism, instead of just some sequences related to genes. These probe sequences usually overlap, being a typical configuration probes with 25 nucleotides, where the first 5 of them overlap with the last ones of the previous probe. Tiling arrays increase by two orders of magnitude the number of probes in a microarray, and they are a revolution in microarray technology, permitting new applications besides gene expression analysis, such as ChIP-on-chip or transcriptome mapping.

2.2.4 *Microarray Applications*

Gene expression profiling, used to compare the expression level of genes among two or more conditions, is still the most widespread use of microarrays. It is mostly used to find groups of genes with the same behavior under certain circumstances, thus identifying the biological processes in which they are involved. Some of the genes in each group may be already classified as related to a certain function, which could include the rest in the function under the assumption of "guilt by association". This kind of analysis has several applications:

- *Agriculture*: for example, a comparison between raw and ripened tomatoes will lead to the detection of genes involved in ripening and therefore to the study of methods to conserve or ripen tomatoes as desired.

- *Pharmacy*: the identification of genes that are regulated by a certain drug potentially provides insight on the action of the drug.

- *Clinical diagnosis*: of relevant diseases is another profitable field for gene profiling. For example, the ability to find cancer cells based on gene expression, or the design of individual therapies based on expression profiling are well underway.

- *Gene mapping*: on a basic scientific level, microarrays have been used to map the cellular, regional or tissue localization of genes and their products. For example, by analyzing samples from different tissues by means of microarrays, one knows which genes are differentially expressed on that tissue.

- *Comparative Genomic Analysis*: is a recent application that tries to characterize the genes within an organism and their functions

4 Density refers to the number of probes in the chip per area unit. The higher the density, the higher the number of probes

by comparison with a reference organism for which the genome is already complete. This is a convenient shortcut to characterize new genomes without the large investment necessary for a traditional genome project.

There are other applications of microarray technology, with slight variations of the methodology and the analysis process:

- *Single Nucleotide Polymorphism (SNP) microarrays*: a SNP is the difference in just one nucleotide between two nucleotide sequences. These microarrays are designed to detect the presence of SNPs between genomic samples, which provides a genetic basis for identifying disease genes, predicting environmental effects and designing personalized treatments.

- *Chromatin InmunoPrecipitation on gene chip (ChIP-on-chip)*: this technique combines chromatin immunoprecipitation with high-density microarray technologies such as tiling arrays in order to, for example, identify binding sites of DNA-binding proteins on a genome wide basis[5].

- *Comparative Genomic Hybridization (CGH)*: is another important application of microarray technology. In this case, gene copy numbers[6] are compared between two samples by using genomic DNA rather than RNA transcripts for the microarray probes. It is usually utilized in tumor analysis to detect gene duplication, amplification or deletion events.

2.2.5 *Microarray Sources*

Regarding microarray building, there are two main sources of microarrays: *in-house* and *company-specific*. Small laboratories usually do not have the technology to create their own microarrays, so they rely on company-specific microarrays to do their researches. Nevertheless, the offer of company-specific microarrays is so large that it is usually enough to most of the laboratories and research centers, including the large ones. However, sometimes there are special needs that require the development of microarrays *ad hoc*. It occurred that a laboratory developed a microarray for a given organism before there were a commercial offer for it, and today both microarray platforms coexist. This is the case of Sanger's[7] in-house microarray for *Schizosaccharomyces pombe* and the Yeast2.0 Affymetrix microarray.

It is generally accepted that companies such as Affymetrix, Agilent and Illumina are the front runners with regards to microarray technology, however there are many more microarray companies out there[8].

5 A DNA-binding protein is a protein that attaches to the DNA sequence at some point. The technique isolates the protein when attached to the DNA, and breaks the DNA to select only the sequences with an attached protein by means of the corresponding antibody (immunoprecipitation). These sequences form the sample to use in the array

6 For example, humans are diploid organisms, so they have two copies of each chromosome, and therefore of each gene. This technique do not detect structural changes such as *trisomy* (three copies of a chromosome), but duplications or deletions of areas within chromosomes, giving a kind of *virtual karyotype*

7 The Sanger Institute is a genome research institute primarily funded by the Wellcome Trust

8 See http://www.nslij-genetics.org/microarray/company.html for a comprehensive list of microarray companies

Each company or laboratory select the DNAsequences that will be in the microarray, depending on their specific purposes, but they usually cover the known genome of the organism, with several different probes per gene. The identification of DNA sequences is also company-dependent, so specific sequence-to-gene mappings are necessary[9].

Regarding microarray retrieval, there are several public repositories. The three most important are maintained by state-funded institutions at the United States, Europe and Japan. Gene Expression Omnibus (GEO) [43] is maintained by the National Center for Biotechnology Information (NCBI) in the USA. ArrayExpress [95] is the European counterpart to GEO, and is maintained by the European Bioinformatics Institute (EBI). GEO and ArrayExpress are the most comprehensive microarray repositories, holding, as to February 2009, 280.000 and 220.000 samples, respectively. Both of them are MIAME compliant (see section 2.3.2) and have evolved to provide additional services, further than simple microarray experiment searches [10, 94]. Gene expression profiles and gene atlas are probably their main secondary contributions. Gene expression profiles are transcription profiles of genes along a curated selection of microarray experiments, presenting a wider look at a gene than its transcription profile on a single experiment. Gene expression atlas goes beyond, integrating several comparable experiments on a single data matrix.

The Center for Information Biology gene EXpression database (CIBEX) [66] is the Japanese public, MIAME-compliant, database for microarray data, but it is considerably smaller that their western counterparts (about 1200 samples). It is also usual to find microarray experiments as supplementary material of related papers [5, 31], but today most of the journals require the publication of experiments on a public repository, so GEO and ArrayExpress remain as the main sources of microarray data.

2.3 BIOLOGICAL KNOWLEDGE

The scientific community whose research resides in the genomics domain is extremely large. Thanks to journals, databases and repositories, the outcomes of these researches are available almost in real time. As a result, our knowledge of biology grows everyday, and does so at a high rate. Microarray experiments are designed based on this available knowledge, in order to increase this knowledge. Microarray data analysis makes use of the available information to validate its results, but also to tighten and simplify the analysis. In this section we will briefly survey some knowledge sources related with the two dimensions of expression matrices: genes and conditions. Both dimensions of knowledge, along with the information about the microarray experiment itself, expand and complement the microarray (see fig. 6).

9 Sequence-to-gene mappings should be updated periodically, because the improvement in the understanding of the genome structure can make some probes to fall out of the new gene sequence limits

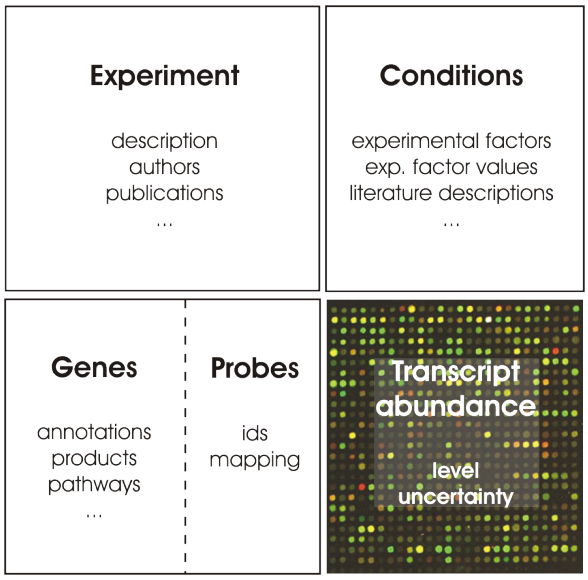


Figure 6: The expression levels are expanded by gene and condition related knowledge, and also by the experiment details.

2.3.1 *Gene-related Knowledge*

Gene and protein-related knowledge is one of the areas with a higher rate of new discoveries in the past years (for example, see figs. 7 and 8). There are several aspects of biological knowledge related to genes, we will grossly summarize them in:

- *Basic information*: name, synonyms, brief description, organism, location, sequence and other general characteristics of the gene. Although not immune to change, it is the most stable gene-related information. However, changes may occur. For example there are lots of genes with unknown functions, approximate locations, etc.

- *Annotations*: they relate genes with different biological concepts. We focus on Gene Ontology (GO) annotations, which link genes to biological terms from a controlled vocabulary (GO terms). GO [8] is currently divided into three ontologies: Biological Process (BP), Molecular Function (MF) and Cellular Component (CC). BP contains terms related to a biological objective to which the gene or gene product contributes, MF defines biochemical activities of a gene product and CC refers to the place in the cell where a gene product is active.

- *External relationships* to other biological concepts, such as gene products (proteins), biological pathways or transcription regulatory networks.

All this information comes from a heterogeneous bunch of sources, and the scientific community makes a gargantuan effort to curate and integrate this information into public repositories and databases. Here are some of them:

- *Basic information*: Entrez Gene from the NCBI is the most important database of genes. Based on *RefSeq* genomes, it provides

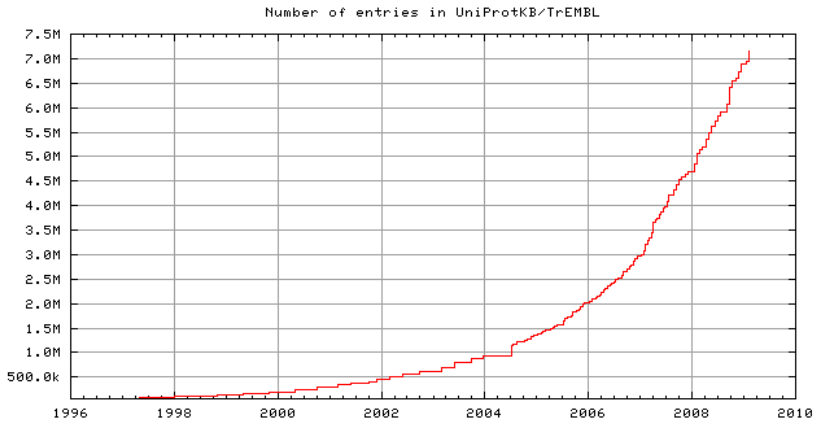


Figure 7: The increase of entries in UniProtKB/TrEMBL clearly follows an exponential rate (found at http://www.ebi.ac.uk/swissprot/sptr_stats)

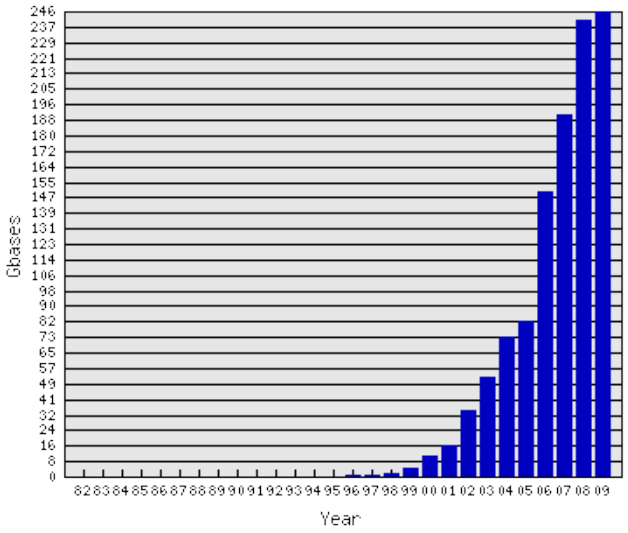


Figure 8: The number of nucleotides added to the Embl sequence database per year, as to February 2009. Note how in just two months, 2009 has surpassed 2008 (found at http://www.ebi.ac.uk/embl/Services/DBStats)

most of the basic information described above. It makes use of several resources from the NCBI, and also from external sources to complete this information. The database is searchable via *http* or by its SOAP API, Entrez Programming Utilities[10].

- *Annotations*: The Genome Annotation Project[11] coordinates the GO annotation of several organisms. This is a huge effort that comprises different groups, for example Gene Ontology Annotation (GOA) for human, Saccharomyces Genome Database (SGD) for yeast and Rat Genome Database (RGD) for rat. Most of them provide programmatic and web query access.

- *External relationships*: UniProt[12] is the main database for proteins, and allows searches by gene, so we can link a gene to its

10 http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html
11 http://wiki.geneontology.org/index.php
12 http://www.uniprot.org/

gene products. The Kyoto Encyclopedia of Genes and Genomes (KEGG) [73] is the best resource for gene mapping to biological pathways, although it is also a general gene repository. BioCarta[13] and Reactome[14] are other examples of biological pathway databases.

The mentioned sources are just some of the enormous number of available databases[15]. Unfortunately, despite the efforts, as of today the database heterogeneity is one of the major obstacles for bioinformaticians. Different sources have different gene identifiers (for example, see table 1), API retrieval formats change with time and different sources may provide different results regarding the same gene information.

SOURCE	GENE ID	COMMENT
Official name	HRAS	Harvey RAt Sarcoma viral oncogene
Synonyms	HRAS1, K-RAS...	Up to 12 synonyms found
NCBI	3265	NCBI id
Ensembl	ENSG00000174775	Ensembl id
UniProt	GTPase HRas	Related protein name
KEGG	K02833	Related protein code for KEGG
Affymetrix	1590_s_at, 35701_at	Probes in Affy chip hgu95av2

Table 1: Heterogeneity of gene identifiers for Homo sapiens oncogene HRAS.

2.3.2 *Condition-related Knowledge*

The information concerning the experimental conditions is much less structured than gene related information. Usually, authors use natural language to describe conditions, these descriptions are only available through publications, and the gene expression matrix only contains arbitrary condition identifiers.

However, fortunately the Microarray Gene Expression Data (MGED) Society has defined a list of essential information that must be present and structured in order to be capable of interpret and replicate a microarray experiment: the Minimal Information About Microarray Experiments (MIAME) [19]. Major microarray public repositories are MIAME-compliant. Regarding the information about conditions, MIAME says that it must be "the essential sample annotation including experimental factors and their values." An Experimental Factor (EF) is a variable of our experiment, for example the studied organism, the part of the organism from which the samples are taken, the age, the sex or the disease state of the organism. An Experimental Factor Value (EFV)

13 http://www.biocarta.com/

14 http://www.reactome.org/

15 See for example ELIXIR, the European infrastructure project for biological information: http://www.elixir-europe.org. This project aims to the identification European biology-related infrastructure projects and to the design of a shared platform to facilitate their access.

is the specific instance of the EF for a given experiment (for example: Homo sapiens, liver, 21, female or control). The current format to distribute a MIAME-compliant microarray experiment is MAGE-TAB [102]. Note that the four data files of the overall structure of MAGE-TAB coincide with the mayor entities involved in a microarray (see fig. 6):

- Investigation Description Format (IDF) describes the experiment.
- Sample and Data Relationship Format (SDRF) relates with experimental conditions.
- Array Design Format (ADF) contains information about the microarray design and probes.
- The data matrix files stores the transcription levels and other related information such as uncertainties or p-values.

3 GENE EXPRESSION DATA ANALYSIS

The Milky Way is nothing else but a mass of innumerable stars planted together in clusters. — Galileo Galilei

Microarray experiments determine the transcript abundance of an organism's genes under different conditions. Gene expression data analysis tries to identify groups of genes that exhibit similar behavior under certain conditions from microarray experiments. In this section we will briefly review the microarray data analysis process, summarizing its different options.

For exhaustive discussions on microarray data analysis, please refer to [16, 61]

3.1 MICROARRAY DATA PRE-PROCESSING

Due to the numerous procedures involved in a microarray experiment, and because of the natural biological variability, microarray data are inherently noisy and have high dimensionality, so it is desirable to carry out the analysis of these data within a statistical framework ([16], chapter 4).

It is usual that during microarray preparation, specially during the image analysis, some of the probe intensities are lost. It is usual to give to these lost intensities (called *missing values*) an estimated value previously to the analysis of the whole data set. In the case of microarray data pre-processing, a number of techniques may be applied, from using zero intensity for missing values to more elaborated estimations, such as the K-Nearest Neighbors (KNN) imputation[1].

Apart from missing values, the whole process of creation of microarrays has several sources of systematic variability: sample preparation, hybridization, scanning and experimenter bias can produce variation on data. The *normalization* process minimizes these variations, although it is not a perfect method. Variability will always exist, and it should be provided with final transcription values, although it is usually ignored. There are three methods of normalization: *total intensity*, *ratio intensity* and *regression*:

- Total intensity normalization applies the same transformation to every probe and sample. A common total intensity transformation is *centralization*, which transforms the data so the mean is zero and the standard deviation is one ([70], page 24).
- Ratio intensity methods take the expression values of one of the samples as the canonical values and normalize the intensities of other samples accordingly (see, for example, the microarray experiment of Chen et al. [31]).
- Regression methods build models to correct the regression curves that fit to intensity levels, and transform them to lines with slope one and intercept zero (see, for example, the LOWESS model [145]).

1 KNN imputation selects the k nearest gene profiles to the gene profile of the missing value, and then sets the mean of these neighbors for the missing value as the estimation.

Finally, note that a probe is not a gene. Although usually probe sequences are selected because they are part of a gene, it is possible to have several probes for each gene (for example, several Affymetrix platforms have 11 probes per gene) or probes not related to a gene (this happens specially in tiling arrays, see section 2.2.3). Note also that the knowledge about genes evolves, so gene-to-sequence mappings may change over time, sometimes discarding or adding probes related to a gene. Anyway, there is a need for a *summarization* step that will comprise probe transcription levels to gene transcription levels, either before or after normalization. Summarization is usually a mean measure or a linear statistical model such as the median polish [137].

3.2 GENE EXPRESSION DATA ANALYSIS

Once the data are pre-processed, it is time to analyze them. Although this thesis focuses on gene expression analysis and the interpretation and representation of its results in the case of biclustering, it is important to know about microarrays (chapter 2) and their pre-processing (section 3.1) in order to detect possible errors or bias in the input data and to provide a (at least, superficial) biological interpretation and validation of the analysis.

At this point, expression data coming from the microarray have become a numerical matrix A, with rows representing genes and columns representing conditions. The number of rows in the expression matrix has as upper limit the number of genes in the genome[2], which means from around 3000 in *Escherichia coli*[3] to approximately 30000 in *Mus musculus* (mouse). The number of conditions depends on the experiment, it can be just two (normal versus disease) or tens or even hundreds (several replications of different stress conditions, tissues or disease states). Therefore, the typical dimension of a gene expression matrix is in $10^{[3-4]} \times 10^{[1-2]}$.

Homo sapiens *genome contains around 25000 genes*

There are many different methods to analyze gene expression data, but despite its number, we can summarize them in *filtering* and *classification* methods. Filtering methods are generally used for hypothesis testing (*does this patient have cancer?*) while classification methods are used for exploratory analysis (*what is the response of bacteria to stress conditions?*). From a wide point of view, exploratory analysis discovers new knowledge and afterwards, hypothesis testing uses that knowledge on practical cases. For example, we need to perform wide exploratory search studies for genes related to brain cancer before we can use these genes as a flag to diagnose brain cancer.

The most usual filtering analysis is to search for genes with transcription levels above or below a certain threshold for a given condition (the gene is *up* or *down-regulated* for the condition). These genes with very high/low transcription levels give way to too many/few proteins of the related types, therefore changing the behavior of the organism. This is enough for several analyses, mainly in biomedicine: if some of the filtered genes in a patient sample coincide with the genes known to be over-regulated for a given disease, this will help to confirm a diagnosis. *Differential analysis of expression* is the most popular filtering method, and consists in the comparison of the gene expression on a determinate

2 Probes from tiling arrays use to be treated "as is", without summarization, increasing dramatically the number of elements, but its analysis is out of the scope of the thesis

3 *E. coli* is a bacterium found in the lower intestine of warm-blooded animals

condition against a control condition, defining expression thresholds based on ratio, mean, variance, etc.

On the other side, classification methods try to characterize the overall structure of the expression matrix, revealing separated groups of genes with similar behaviors. This can be done with the help of available knowledge (*supervised* classification) or not (*non-supervised* classification). Supervised classification mixes knowledge discovery with the use of available knowledge, so it is less useful for raw exploratory analysis but usually gives more accurate results. Supervised classification is also called class based prediction, discriminatory analysis or supervised learning. On the other hand, non-supervised classification does not require any additional input apart from the raw data, but the results are usually less precise. Non-supervised classification is also called automatic prediction, clustering analysis, partitioning, grouping, data segmentation and non-supervised learning. It is also possible to find non-supervised techniques that incorporate available knowledge to guide the analysis, but keeping part of the capability of discovery of non-supervised classification [103].

We will focus on classification methods because exploratory analysis is a more open field than hypothesis testing and it also includes it somehow[4]. For an exhaustive study of classification methods, refer to [16] (chapters 7–18). Here we provide a brief enumeration of supervised methods:

- *Discriminant analysis*'s objective is to find the combination of conditions which best separate two or more classes of genes (or viceversa). Depending on the characteristics of the combination, we can have Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Diagonal Linear Discriminant Analysis (DLDA), etc.

- *Nearest neighbors* are among the simplest methods of supervised learning. The most spread method is the KNN algorithm. In this algorithm, for each gene, its k nearest neighbors are found given a distance measure, and the gene is assigned to the most common class among these neighbors.

- *Support Vector Machines (SVM)* give a geometrical solution to the classification problem. Given two classes, SVM searches for the hyperplane that better separates them, maximizing the distance between the hyperplane and the closer individuals of each class.

- *Decision trees* are hierarchical structures where the leaves represent classifications and branches are conjunctions of features that lead to these classifications.

- *Artificial Neural Networks (ANN)* can be used for classification, defining a network with as many input nodes as rows in the matrix and as many output nodes as classes[5]. An arbitrary number of hidden nodes are also defined, and the neural network is trained until a classification is obtained. Several variations of the architecture and the learning model have been proposed in the literature [15, 86].

4 Often, exploratory analysis is a succession of hypothesis testings, and some authors model it like that (for example see Keim et al. analytical process [76] in section 5.1)

5 Note that, in several supervised methods, the number of classes in the data is known a priori

And of non-supervised methods:

- *Clustering* groups genes with similar profiles under all the conditions. A further description of clustering is in section 3.3 below.
- *Biclustering* groups genes with similar profiles under certain conditions. As the focus of this thesis is biclustering for gene expression analysis, it is reviewed in detail on chapter 6.
- *Self-Organizing Maps (SOM)* are a non-supervised type of ANN that implements competitive learning: neurons must compete to fit to a given input, redefining the weight patterns of the network, or even its structure. An example of use in gene expression analysis can be found in [125]. Some authors consider them just another kind of clustering [29].
- Principal Component Analysis (PCA)/Singular Value Decomposition (SVD): these methods search for the main features that separate data into groups, by means of dimensionality reduction on principal components (PCA) or eigenvectors (SVD). This reduction of dimensionality is a good initial approach to tackle complex data, and it is sometimes used as a previous step for clustering. On the other side they can oversimplify the data set, and they are very sensitive to pre-processing.

There is no consensus about which is the best classification method ([16], cap. 1). There are methods that apply better to some problems than others, and some authors have reported that complex methods do not perform better than simple ones [39, 38]. In the field of microarray data analysis, non-supervised clustering was satisfactorily used a decade ago [44] but now, with the growth of available genetic knowledge, methods tend to incorporate this knowledge to give more accurate outputs [103]. This way, unsupervised methods tend to become "biologically supervised" methods.

Generically, non-supervised methods output more results than supervised methods. Some methods can overwhelm the analyst with results, making the analysis almost as complicated as to inspect raw data directly. To avoid that, it is usual to simplify the output by using thresholds or, for example, by using biological knowledge. Although this is a valid approach, it will always reduce the exploratory capability of the analysis techniques. One of the objectives of this thesis is to discuss how to visualize a relatively large number of groups in order to make its exploration easier and to help on their interpretation.

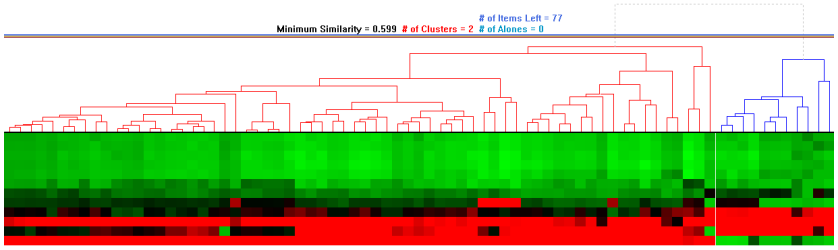
3.3 CLUSTERING

Clustering may be defined as a process that aims to find partitions or groups of similar objects, called *clusters*. In a genomic expression application, a cluster may consist of a number of genes whose expression patterns are more similar to genes within the same cluster than to genes within other clusters[6]. Clustering has become a fundamental approach to analyze genomic expression data. Since the pioneer use of Eisen et al. [44], it has provided the basis for novel clinical diagnostic studies and other applications (just some examples of it are [5, 31, 136]).

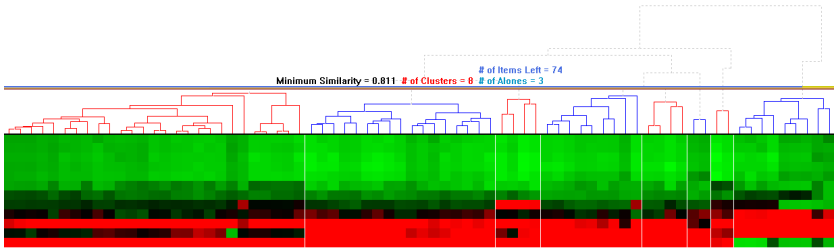
6 Analogously, clustering may also be applied to conditions.

Typical clustering algorithms are based on the optimization of a partitioning quality measure. Generally these measures are related to the heterogeneity of clusters (*compactness*) and their separation from the rest of data (*isolation*). Thus, a basic clustering approach aims to search for a partition that minimize intra-cluster distances and maximize inter-cluster distances. There are several types of metrics to assess these distances, typically based on the sum or manipulation of distances among elements [70].

There are two major types of clustering systems: *hierarchical clustering* and techniques based on *iterative relocation*. Some authors classify other types or non-supervised techniques such as SOM or biclustering as other kinds of clustering too. *Hierarchical clustering* is perhaps the best known clustering method for expression data analysis. The main objective of this technique is to produce a tree-like structure in which the nodes represent subsets of an expression data set. Thus, individual genes (the leaves of the tree) are joined to form groups, which are further joined until a single group is obtained. Afterwards, a cut threshold is defined on a certain level of the tree, taking the branches at this point as groups (see fig. 9). How to define the threshold is non trivial, and it is usually left to the criterium of the analyst, possibly with the help of inter and intra cluster measures or validation indices (see section 6.5).



(a) Hierarchical clustering cut with two clusters



(b) Two clusters (red and blue) obtained by hierarchical clustering

Figure 9: Clusters at different thresholds of a hierarchical clustering. Under the tree (its representation is called *dendrogram*), a heatmap represents expression levels (see chapter 7). Clusters are separated by white lines that correspond to the blue/red branches. Figures generated with [114].

Iterative relocation methods involve a number of "learning" steps to search for an optimal partition of samples. Such processes require the specification of an initial partition or some knowledge on the underlying class structure, such as the number of groups in the data. The most common techniques in this category are the *k-means algorithms*. The k-means method categorizes samples into a fixed number k of clusters, but

it requires a priori knowledge on the number of clusters representing the expression data under study.

Once a clustering algorithm has been selected and applied, analysts face questions such as *which is the best partition?* or *what is the right number of clusters?* To answer these questions it is required to use estimations based on validity indices (section 6.5 describes the types of validity indices for clustering and biclustering).

4 INFORMATION VISUALIZATION

A picture is worth a thousand words. An interface is worth a thousand pictures. — Ben Shneiderman

Information visualization is related to the use of interactive visual representations of abstract data to amplify cognition [141]. The more complex the abstract data are, the more important information visualization is to understand them. Today technologies generate very large amounts of data, with complex or unknown structure, in almost every research field. Astronomy, biology, physics, sociology, etc. share the need for analysis of large, complex information. All these disciplines make use of numerical analysis methods such as the ones described in chapter 3, but the amount of information makes it difficult to understand, interpret or represent their results. Numerical analysis exploits our abstract (logical, mathematical) intelligence in order to understand problems. However, to understand a problem, we make use of other kinds of intelligence, specially verbal intelligence and visual intelligence. Information visualization exploits visual intelligence in order to help abstract intelligence.

Howard Gardner introduced the theory of multiple intelligences in 1983

In this section we will briefly summarize the origins of information visualization and review how information visualization has been applied to bioinformatics, specially to microarray data and clustering.

4.1 ORIGINS AND RELATED FIELDS

Over history, information has been visualized to aid thinking. Pictures, maps, diagrams, etc. are and have been used to represent concepts, and several disciplines have appeared to achieve it: cartography, information design, statistical data graphics, etc. With the improvement of technologies and the increase in the number and complexity of data, these disciplines evolved, being information visualization the latest instance.

Older disciplines used static diagrams, progressively utilizing color and text to clarify them. A key step in visualization disciplines is the use of characteristics such as color, width or location to convey abstract aspects of data. Although already used in ancient cultures such as the Greek and Chinese (figs. 10a and b), it was after Descartes and the Scientific Revolution at the beginning of 17th century that this characteristic was widespread (fig. 10c). Charles Minard's illustration of Napoleon's Russian campaign is an excellent example that has become a classic in the field (fig. 11).

All these presentation graphics illustrate a concept already known by the designer, however information visualizations are designed to uncover unknown phenomena. Presentation graphics tend to be static and normally in printed format, while information visualizations are computerized and interactive. Apart from this, both types of graphic essentially share the same principles related to perception and visual intelligence.

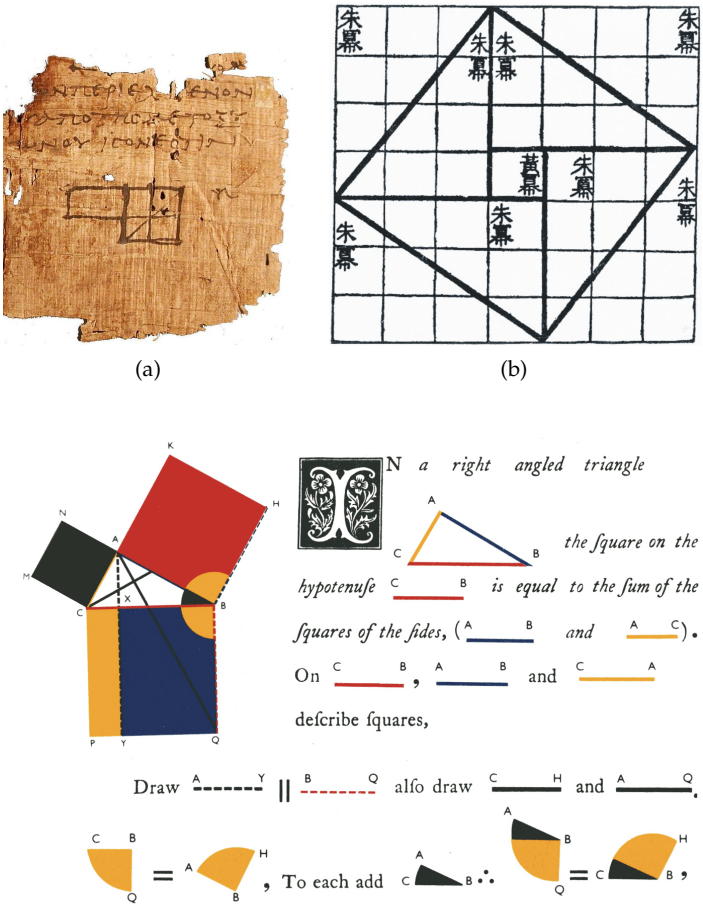


Figure 10: a) Detail of a diagram in one of the oldest fragments of Euclid's book *Elements*, dated circa AD 100. b) Proof of the Pythagoras' Theorem just by means of a diagram, found in a classical Chinese mathematics book dated circa AD 200. c) Oliver Byrne's paragraph explaining Pythagoras' Theorem in his book *The Elements of Euclid* (1847). The use of colors and the integration of figures and text speed recognition and linkage between diagram and proof (figure found in [133]).

Scientific data visualization is an area close to information visualization. Although relevant authors [28] define scientific data visualization as a representation of data in relation with a physical space and oriented to area specialists, while information visualization has a more abstract nature and general audience; the fact is that they overlap to a high degree and they often respond to the generic label of *data visualization*.

Psychology branches related to perception are also associated to information visualization for an obvious purpose: to exploit the properties of human perception in order to improve the effectiveness of a visualization technique. Human-Computer Interaction (HCI) is concerned with the study and design of human-centric interactive computer systems,

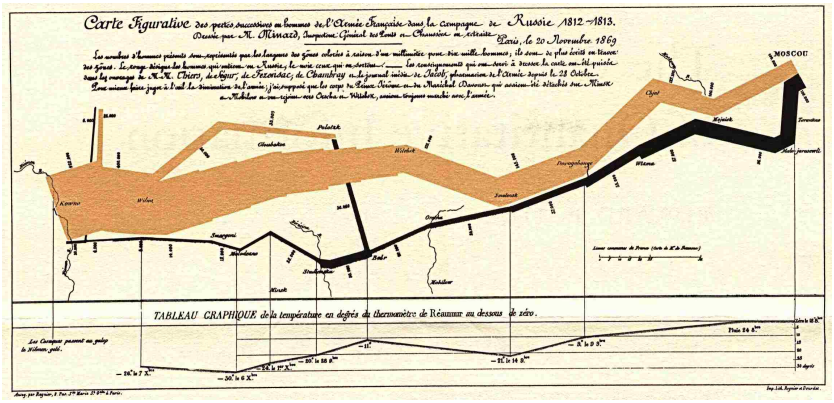


Figure 11: Minard's illustration of Napoleon's Russian campaign. It conveys the size of Napoleon's army by the width of the band, using color to convey advance (light brown) and retreat (black). Small branches represent tactical movements of soldiers. A lower line graph represents temperature during the retreat. Figure found in [132].

so information visualization can be regarded as a parallel field, since visualizations are part of these systems[1].

4.2 PRINCIPLES OF INFORMATION VISUALIZATION

This section reviews the main aspects of information visualization in an practical way as they relate to our visualization objectives. For more theoretical and exhaustive compilations please refer to the comprehensive books on the theme [141, 28].

4.2.1 *Interface design*

The most widely used guideline for information design is the *Visual Information-Seeking Mantra* [120]:

> Overview first, zoom and filter, then details on demand

The mantra can be applied to several types of data, and is divided into the following tasks:

- *Overview*: gain an overall impression of the entire collection.
- *Zoom*: focus on items of interest.
- *Filter*: get rid of uninteresting items.
- *Details-on-demand*: select an item or group and get the related low-level information when needed.
- *Relate*: view relationships among items.
- *History*: keep a record of actions to support undo, replay and progressive refinement.
- *Extract*: allow to export sub-collections and query parameters used.

1 Not surprisingly, relevant authors on information visualization are also related to HCI, such as Ben Shneiderman, founding director of the Human-Computer Interaction Lab (HCIL)

The data on which the mantra is applied can be of several types, enumerated below. Table 2 shows biological examples of each type.

- *1-dimensional*: data organized in a sequential order.
- *2-dimensional*: planar data, such as geographical maps.
- *3-dimensional*: real world objects such as molecules or buildings, or abstract data designed in a 3-dimensional structure.
- *n-dimensional*: data with $n > 3$ variables. It is the case of relational databases but also of most of the matrices.
- *Temporal*: data with start and end times such as processes or historical events.
- *Trees*: data where each item is linked to one parent item.
- *Networks*: data where each item is linked to an arbitrary number of other items.

DATA TYPE	EXAMPLES IN BIOLOGY
1-dimension	Nucleotide and amino acid sequences
2-dimension	2D MRI, microarrays
3-dimension	Protein structures, 3D MRI
n-dimension	Gene expression matrices
Temporal	Experimental conditions through time
Trees	Phylogenies, clustering trees
Networks	Biological pathways, PPI networks

Table 2: Data types and biological examples.

4.2.2 *Data Representation*

Apart from dimensionality and relationship nature, the most important thing to think about data is the *level of measurement*. According to S.S. Stevens [123], there are four levels (nominal, ordinal, interval and ratio), but they are usually reduced to three, greatly influenced by the demands of computer programming:

- *Category data*: also called nominal data, they are basically labels. For example, gene names are nominal.
- *Integer data*: they can have a discrete number of values where order is important. A discretized gene expression matrix or a nucleotide sequence have an integer nature.
- *Real data*: they can have any number of values, including zero or negative values. Gene expression and p-values are examples of real data.

The p-value is a measure of probability ranging from zero to one often used in statistical significance tests (see sec 6.5)

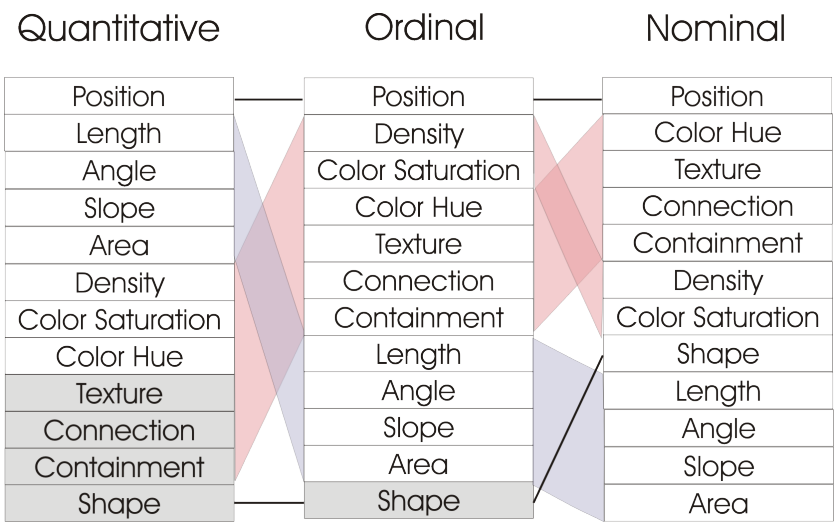


Figure 12: Ranking of perceptual effectiveness to measure different data types with the corresponding visual variables. Shadowed variables are not relevant to the corresponding data type (reproduced from [83]).

The scale of measurement is very useful in discussing visualization techniques. For example, using shape to convey category data can be misleading because we tend to interpret size as representing quantity [141]. Mackinlay [83] ranks the accuracy with which human perception interpret visual variables as measures (see fig. 12). The higher in the rank, the better fitted to measure the corresponding kind of data. Position is the best visual feature under any circumstance, so it should be used to represent the most relevant aspect of the data. On the contrary, shape is ineffective, and must be reserved for nominal data. Dimensional characteristics (length, angle, slope and area) are good to represent quantitative information. Color, texture and relationships are better fitted for non-quantitative data. Mackinlay's ranking is good as a reference, but it is important to know that it lacks of some visual

encodings such as motion (rotation, speed, direction, flicker), curvature or convexity/concavity.

In addition to the ranking about the accuracy of perception, we can take the criterion of immediacy of perception. The quickest perceptual reaction is *preattentive processing*, the mechanism by which certain encodings are easy to be visually identified, even after a very brief exposure (typically around 100 milliseconds). A special characteristic of preattentive processing is that the time to identify preattentively distinct objects is independent to the number of distracters (irrelevant objects). Almost any visual characteristic can be preattentively distinguished (for example, a red item among black items; or a triangle among several circles). However, preattentive symbols become less distinct as the variety of distracters increases [141] (for example, a red item is less preattentive among items with several different colors).

After preattention, the *visual working memory* holds the visual objects of immediate attention. Its capacity is limited to a small number of simple visual objects or patterns (around three) and positions (about nine, but only three linked to visual objects). It is separate from verbal memory, and to store the visual objects in long term memory requires a semantic encoding (for example, by means of labels attached to the objects). The visual thinking is performed with this reduced set of visual objects, which is constantly changed within a loop of eye-movements to switch attention to other objects. It is also supported by:

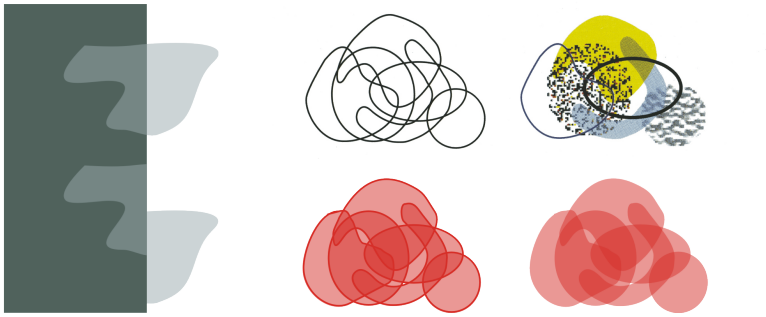
- The temporary link of visual objects with verbal-propositional information (the grouping is called an *object file* [71]).
- The recall of visual scenes or contexts from long-term memory (*gists*) because they are familiar to the visual objects we are inspecting.
- The grouping of several simple concepts into a single complex one (this process is called *chunking*).

Therefore, data representation can favor the visual thinking process by means of simple object representations (to minimize visual memory load), semantic marks (to favor the creation of object files), familiar environments (interfaces already known by the user) and grouping laws (such as the Gestalt laws, see section 4.2.4).

If the data representation is good enough, *postattentive processing* [142] (the mechanism by which a visual representation persists after attention changes to something else) can take profit of visual working memory while switching the focus of attention during the navigation through the visualization. Otherwise, if the scene and the visual objects are complex, the user will need much more changes of attention to go back and revisit visual items. An example of the limits of visual working memory is the *change blindness* [89], where people become blind to substantial changes between two images if these changes do not draw their attention.

4.2.3 *Transparency*

Transparency is an interesting visual encoding when we think on the representation of overlapped entities, such as biclusters. In these cases it is desirable to convey overlap on a layered form but there are many perceptual pitfalls on it [141]. One of these pitfalls is that transparency is perceived only when good continuity is present (see fig. 13a). Another one is the interference among different layers, so it is very important to keep them separated on different visual channels such as color, shape, size, texture, motion or depth (see, for example, the difference between the top two representations in fig. 13b). If the overlapped data refer to the same entities (for example, biclusters overlapping one another) and there is no possibility to change the visual channel, transparency can be quantified when up to five items overlap [45] (for example, the two bottom representations of fig. 13b successfully convey overlaps of up to three groups). For a larger number of overlapped items, transparency can only give us a qualitative idea of overlap, so ancillary visualizations are needed in order to quantify it.



(a) (b)

Figure 13: a) Overlapped continuous objects (top) are perceived as transparent. It does not occur with discontinuous objects (bottom) (reproduced from [141]). b) Different ways of conveying overlap: contours and textures (top, reproduced from [141]) and single color transparency, with and without contours (bottom).

The use of color and texture in combination with transparency to convey complex relationships could slightly increase the number of identifiable overlapped items (fig. 13b, top-right). However, there are again issues with the use of a large number of different textures and colors: texture and color cluttering will quickly arise. To summarize, transparency is good at conveying overlapping relationships among objects if they come from different visual channels (such as groups and the elements within, that can be represented with very different encoding) and if there are a low number of objects or they are backed up by other visual encodings.

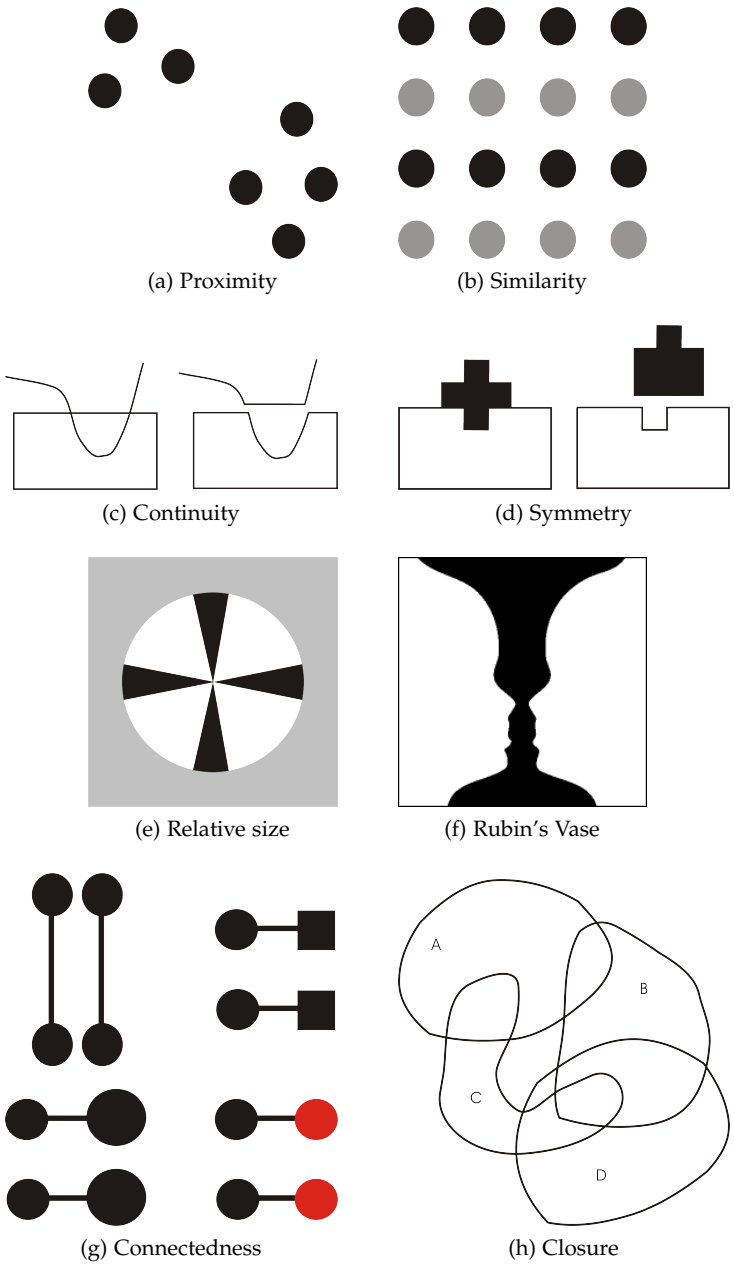
4.2.4 *Group Representation*

Gestalt is the German word for shape, *it is used in English to refer to a concept of* wholeness

The perception of groups is related to how visual entities interact. The perceptual effectiveness of interactions is ruled by the Gestalt laws [30]. Here is a selection of the most relevant Gestalt laws (some examples in fig. 14):

- *Proximity*: Things that are close together are perceptually grouped together. Position is the most relevant visual encoding for data (see fig. 12), and it is also a powerful way to visualize groups.
- *Similarity*: Similar elements tend to be grouped together. It can refer to color, shape, texture or any other separable visual dimension.
- *Connectedness*: Connected entities tend to be grouped together. This is a powerful principle, stronger than proximity, color, size or shape [93].
- *Continuity*: It is more likely to construct visual entities out of visual elements that are smooth and continuous than out of elements that contain abrupt changes in direction.
- *Symmetry*: Symmetrical objects or symmetrical arrangement of objects are more likely to be perceived as a whole.
- *Closure*: A closed contour tends to be seen as an object. This is, presumably, the reason why Venn and Euler diagrams are so powerful for displaying interrelationships among sets of elements (see section 8.1).
- *Relative size*: Smaller areas tend to be perceived as objects.
- *Figure and ground*: The capability to perceive figures as opposed to ground depends on the other Gestalt laws. If figure and ground compete on these laws, the result is ambiguity (such as in the case of the Rubin's Vase, see fig. 14f).

Visualizations usually employ combinations of the Gestalt laws in order to represent groups, and the laws will be very important for the design of clustering and biclustering visualizations.



(a) Proximity (b) Similarity

(c) Continuity (d) Symmetry

(e) Relative size (f) Rubin's Vase

(g) Connectedness (h) Closure

Figure 14: Gestalt laws examples. a) Two groups perceived because of proximity. b) Row perception predominates due to different colors. c) We perceive a curve and a rectangle instead of the more complex figures on the right. d) We perceive a cross over a rectangle (left) easier than an asymmetric shape under the rectangle (right). e) We perceive four black areas within a white circle rather that four white areas within a black circle. f) It is hard to decide if we see two faces on black background or a cup in white background. g) Connection is more powerful than proximity, shape, size or color. h) An Euler diagram is a good example of closure, with clear relationships among sets.

4.2.5 *Data Interaction*

Without data interaction, visualizations are just static graphical representations. The exploration of data from the overview to revealing patterns, subsets or details is mandatory in order to implement an interface design such as the one described in section 4.2.1. Ware [141] identifies three loops of data interaction:

- *Data selection and manipulation*: this loop is focused on how we process visual signals and send decisions to the visualization. It is directly related to the eye-hand coordination, such as reaching an object on the screen by means of the mouse, tracing lines or watching for relevant patterns.
- *Data exploration and navigation*: this loop focuses on a wider point of view which deals with the location and orientation on large visual spaces, the scaling and the rapid interaction with data.
- *Problem solving*: this is the highest level loop, involving how we use the visualizations in order to unveil relevant information and to support reasoning. This level is highly related with visual analytics (see chapter 5).

The simple interactions related to data selection and manipulation frequently takes less than one second, providing that stimulus-response compatibility (for example, to move the mouse to the right makes the cursor to go to the right) is respected. In addition, learning can improve time performance: the more an interface is used, the faster the user manipulates it. It is necessary to provide quick and clear feedback to each user's action in order to make the learning easier.

Because information visualization usually deals with large and complex data, it is usual to display them on large visual spaces. Therefore data exploration and navigation techniques should be carefully designed. This is specially important on 3D and virtual reality environments, but we will focus on two aspects closer to our needs: scaling and rapid interaction.

Scaling tries to solve the *focus-context problem*: to find detail in a larger context. Data scales may be in the context of *space* (for example on a map possible scales are: meters, kilometers and thousands of kilometers); *structure* (provinces, countries, continents); or *time* (seconds, hours, years). Despite the context, all of them are finally displayed on a screen, so the scaling techniques are the same for all of them:

- *Distortion* techniques give more room to designated regions or elements, and decrease the space given to the remaining regions. They provide focus but at the same time keep context, although sometimes the reduction of the context is so large that its structure is lost [141]. The distortion can be multi-foci, expanding different areas at once. *Bifocal distortion* gives a larger size to the selected items and a smaller size to the rest (applied, for example, on [23]). *Fish-eye distortion* [109] gives the largest size to the focused region and progressively reduces the size of objects that are away (see fig. 15).
- *Rapid zooming* techniques switch quickly from detail to overview. Focus and context are not available at the same time, but the change is quick and smooth enough to allow the user to integrate

both scales. This is, for example, the case of Google Earth®[2] or the zooming technique implemented by Prefuse [58].

- In *Elision* techniques, parts of the structure are hidden until they are needed. Sometimes these techniques are called *semantic zoom* [24], meaning that less and less detail is shown as the distance to the focus of interest (or the scale) increases.
- *Multiple windows* techniques are common, specially in mapping systems, displaying one window that shows an overview of the whole visual space and one (or more) that shows expanded details.

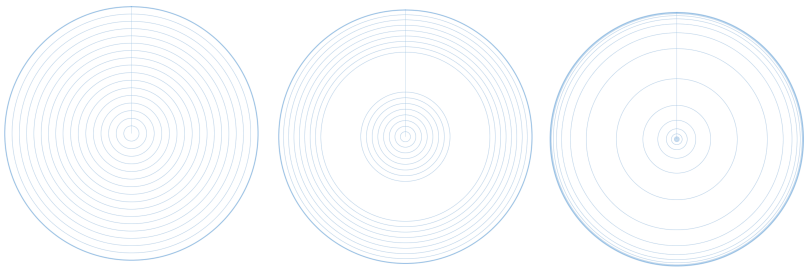


Figure 15: Concentric rings with equivalent widths (left). Bifocal distortion of a middle ring, the remaining rings get an equivalent reduced width (center). Fish-eye distortion centered on a middle ring, closer rings are also magnified and, progressively, further nodes are reduced (right).

In a data exploration interface, it is important that the mapping between the data and its visual representation be fluid and dynamic. As a rule of thumb visual feedback should be provided within 0.1 seconds for people to feel that they are in direct control of the data. A technique that requires rapid interaction with data is *dynamic querying* [4], which performs data filter queries and visualizes the results in real time. Another interactive technique is called *brushing* [12], which enables the highlighting of elements in complex representations. Brushing is specially used on a visualization technique called *parallel coordinates* [68] (covered in detail in section 7.1.2).

The main objective of data interaction is to make the interface *fluid* and *transparent* by 1) supporting eye-hand coordination, 2) using well-chosen interaction metaphors and 3) providing rapid and consistent feedback. Transparency improves with practice, but the interface designer should get to a compromise between interface complexity and discovery capabilities of the visualization. Very simple interfaces cannot find the subtle information within complex data, while very complex interfaces could make the user to give up, or to focus more in the interface than in the problem to solve with it.

To conclude, interaction is a key characteristic of information visualization that converts static representations into functional visualizations.

2 http://earth.google.es

It is sometimes hard to demonstrate its relevance in static media such as paper publications, but the proper design of interaction techniques determines to a large extent the utility of a visualization.

4.2.6 *Multiple-linked Views*

There is evidence that the highlighting of the same data items on different views will improve the proficiency of knowledge acquisition tasks [122]. Making the connection for the user reduces the cognitive load of switching from one view to another. In addition, our capability to detect changes even in peripheral areas of vision (such as visualizations outside of our main focus) allows making observations that would not be possible from two separate views.

Multiple-linked views have been in use since long time ago, in different scientific contexts. Statisticians are very used to handle multiple scatterplots to analyze multidimensional data (see fig. 16). The use of different visualizations (not only several copies of the same visualization) has also been addressed with success in, for example, data simulation [37] or biology [114]. See section 9.2 for some examples of multiple-linked views in bioinformatics.

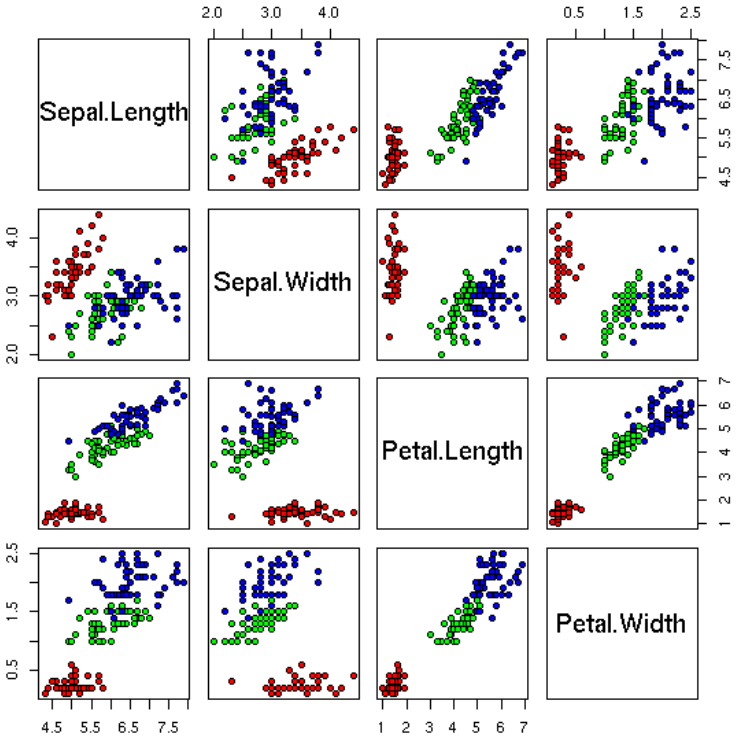


Figure 16: Scatterplot matrix representing two-by-two relationships among the values of four different dimensions of data. Generated with R [65]

.

In addition, the study of the interaction among several visualizations has been addressed since more than a decade ago [25]. The selection of elements within a visualization must generate a query for selection

of elements on other visualizations. Depending on the nature of the visualizations and the data they represent, the complexity of the queries varies. It is also necessary a visual code to make clear that two or more different represent, such as a determinate color or shape.

5 VISUAL ANALYTICS

Over time and cultures, the most robust and most effective form of communication is the creation of a powerful narrative. — Howard Gardner

The emerging field of visual analytics focuses on handling massive, heterogenous, and dynamic volumes of information by integrating human judgement by means of visual representations and interaction techniques in the analysis process [76]. It combines several research areas such as information visualization, data mining and statistics.

Although visual analytics science was born to analyze security threats or disasters [130], it can be applied to any field that involves an analytical process. In particular, bioinformatics is not beyond the scope of visual analytics [110, 107].

5.1 THE ANALYTICAL REASONING PROCESS

This science of analytical reasoning provides the reasoning framework upon which one can build visual analytics technologies. It is a scientific standard probably since Descartes' *Discourse on the Method* in the 17th century.

The analytical reasoning process is structured and disciplined. It is also inherently *iterative*: the process of reaching a judgment usually requires of several iterations or approaches to the problem. In addition, obtaining an answer often produces several more questions, leading to additional analyses about a larger issue.

This analytical process is the basis for the ongoing dialogue between analysts and their information, so the mission of visual analytics is to enable this discourse. This dialogue is called the *analytical discourse*.

The analytical reasoning process can be separated into four major steps [130] (see fig. 17):

1. *Gather Information*: to collect the relevant data to start the analytical process is not trivial[1]. Substeps such as acquisition, parsing, filtering and mining are all included into the gather information phase. Computer science, mathematics, statistics and data mining are disciplines usually involved in it.

2. *Re-represent*: the data acquired in the previous step are now represented in new ways in order to facilitate analysis. Graphic design, information visualization, and HCI come into scene.

3. *Develop Insight*: the analyst interacts or otherwise manipulates the representation. Again, information visualization and HCI are important, along with the relevant knowledge of the analyst in the target field of research.

4. *Produce Results*: after gaining a deeper knowledge on the studied issue through the previous phases, the analyst produces results,

1 Consider, for example, the difficulty of gathering all the information available about a gene because of the heterogeneity of identifiers (section 2.3.1)

either concrete (images, tables, selection of relevant features) or abstract (decisions to perform addition experiments, changes of criteria, etc.). The target research field is the main related discipline (for example genetics, astronomy, medicine or more specific subfields).

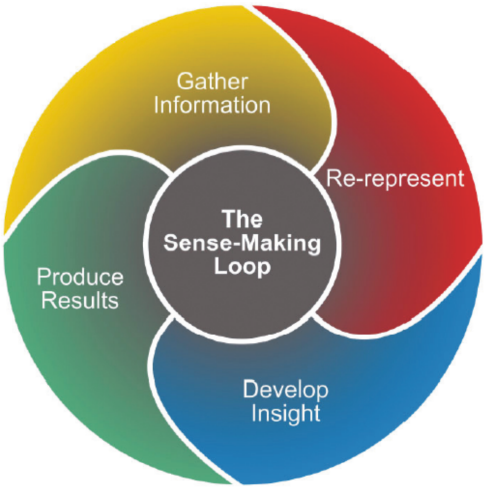


Figure 17: The analytical reasoning process (found in [130]).

Visual analytics covers all these steps, integrating information visualization as part of the process, but also comprising the application of automated analysis before and after the interactive visual representation [76].

Regarding the first three steps, Ben Fry proposed the *Computational Information Design* [49], a cyclic process with several possible iterations. Fig. 18 shows this process, where we can see, for example, that the interaction with the visualizations may lead to refinements in the representation, or to modify statistical parameters in order to inspect a different characteristic of our data. Ideally, these iterations can be done within the frame of a single tool, but usually the analytical process is large, involving several sources, so some of the iterations (specially the large loops such as "represent to acquire flux" in fig. 18) often require to switch to another tool, or to combine different tools. These bridges among resources (databases, web services, software, etc.) should be avoided when possible, because they usually require an additional load in time and effort, involving the change of formats, possible mismatches on identifiers and data structures, etc.

Keim et al. [76] provide a formal definition of the visual analytics process based on entities rather than on steps (see fig. 19). These entities are data sets (S), visualizations (V), hypothesis (H) and insight (I). Data sets are the source for visualizations (V_S), but they can also generate hypothesis by themselves (H_S). Data gathering and parsing is an iterative process involving data (D_W). Visualization facilitates the generation of new hypothesis (H_V), but also hypothesis can be visualized or give way to visualizations (V_H). The user communicates with the visual analytics framework by interacting with the visualizations (U_V, for example,

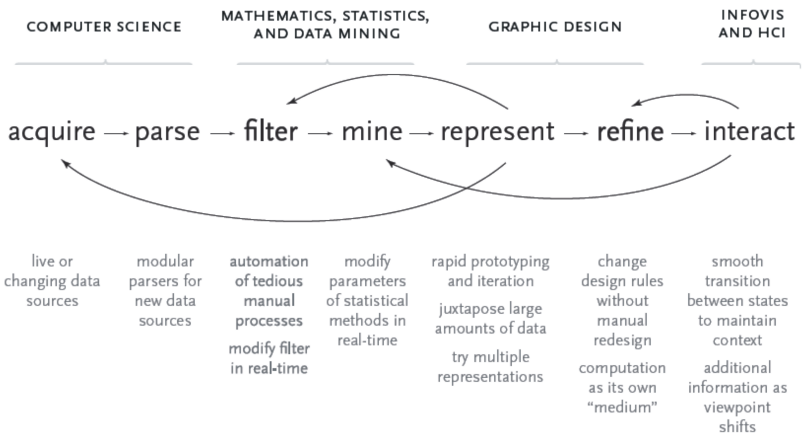


Figure 18: The Computational Information Design (reproduced from [49]).

zooming or selecting) or formulating new hypotheses (U_H). Finally, the insight that leads to decision making is the conclusion of the analyst from the inspection of visualizations (U_{CV}) and the confirmation or refusal of hypothesis (U_{CH}). The achieved insight will serve as feedback for the next step of the visual analytics process.

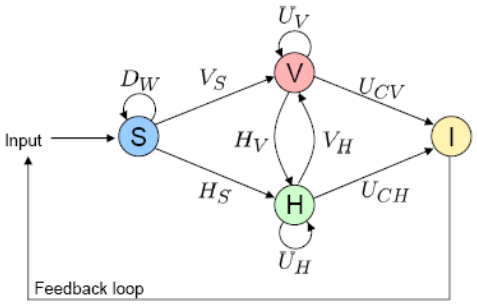


Figure 19: The Visual Analytics Process by Keim et al. [76].

5.2 PRODUCTION, PRESENTATION AND DISSEMINATION

The first three steps of the overall analytical reasoning process (fig. 17) are usually well covered. However, technologists too often overlook the presentation of the results and how do we get to them. This is vital because the analytical process is inherently collaborative. The National Visualization and Analytics Center (NVAC) gives some recommendations in order to improve the capabilities for production, presentation and dissemination of analytical methods and tools [130]:

- Few scientific methods or tools support the creation of an final product (user's guides, developer's instructions, etc.). It is important to develop methodologies and tools that facilitate their use by third parties.
- It is key to develop tools that not only communicate results, but the reasoning that concluded with that results. It should

use appropriate visual metaphors and accepted principles of reasoning and graphical representation.

- Create visual analytics data structures, intermediate representations and outputs that support integration with other broadly accepted tools in the research area, so the need for data acquisition and formatting is minimized.

Several interesting tools are disregarded by the scientific community due to the complexity of use, either by lack of documentation, difficulty of formats, fatal bugs, etc. The focus of a researcher is to develop a good methodology or tool for his objectives, and maybe its publication. The use by third parties is rarely considered, however it is very important to dedicate time and effort in this last step.

5.3 EVALUATION METHODOLOGIES FOR VISUAL ANALYTICS

There are several benefits in incorporating evaluation as part of a research program: verify research hypothesis, encourage research and challenge researchers in a particular area, increase communication, compare technical approaches, etc. The scope of the evaluation can be a single visual analytics approach or a whole research area field. In the second case, international contests and competitions are the usual approach. An example is the InfoVis Contest[2], born as part of the Institute of Electrical and Electronics Engineers (IEEE) Symposium on Information Visualization (InfoVis), which attempts to create an Information Visualization Repository that contains resources to improve the evaluation of information visualization techniques and systems. Furthermore, a Visual Analytics Science and Technology (VAST) Contest branched from the InfoVis Contest in 2006, explicitly citing the evaluation methodology proposed by the NVAC. Finally, another example, focused on data mining, is the Knowledge Discovery and Data mining (KDD) Cup[3], an annual competition of the Association of Computing Machinery (ACM).

Visual analytics systems are complex and require evaluation efforts targeted at different levels. NVAC proposes to consider three levels: component, system, and work environment (fig. 20). The *component level* comprises analysis algorithms and visualization techniques. Algorithms do not require interaction metrics, but other indices usually easy to compute or observe, such as speed, accuracy or limitations. Visualization techniques (either interface designs, interactions or representations) require empirical user observation, including metrics such as effectiveness (time to complete simple tasks), efficiency (number of errors or incomplete tasks) and overall user satisfaction. Some metrics for visualization can be computable, for example the number of objects that can be visualized at once.

At the *systems level*, metrics need to address the learnability and usability of the system, along with the user satisfaction. In consonance with Keim et al.'s cycle (fig. 19), a new measurement approach is the insight-based evaluation. The IEEE InfoVis contests [98] ask the contestants to report on insights gained from the data sets. North [88] characterizes insight as complex and not exact, but deep and relevant. He identifies two mayor methods to measure insight, with and without

2 http://www.cs.umd.edu/hcil/InfovisRepository
3 http://www.kdnuggets.com/datasets/kddcup.html

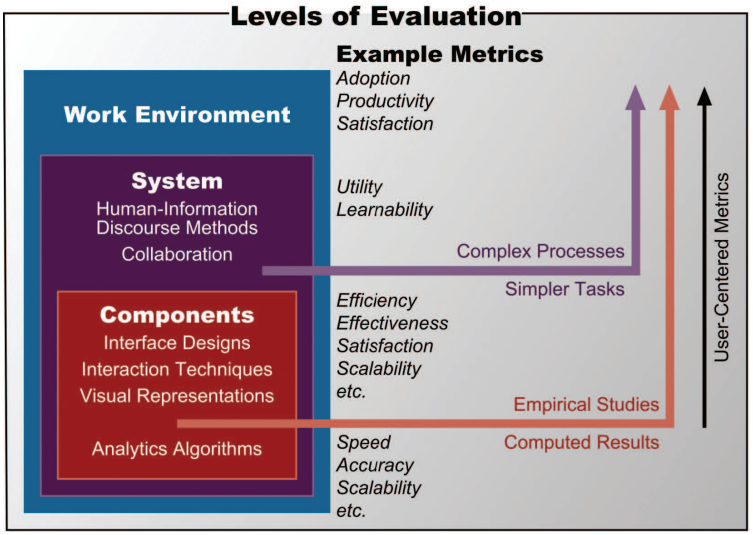


Figure 20: The three levels of evaluation and example metrics (found in [130]).

benchmark tasks. Benchmark tasks are strict, involving predefined tests and answers, so they are not very good to characterize insight, but on the other side they are accurate. Although qualitative, an open-ended protocol without benchmark tasks is also interesting, because it allows the user to verbalize their findings so that evaluators can capture the user's insights. [108] is a good example of a non benchmark-dependent validation of some tools for microarray data analysis by means of clustering.

Finally, at the *work environment level*, the evaluation focuses on the technology adoption, with metrics such as adoption rate, trust and productivity.

Part III

STATE OF THE ART

6 BICLUSTERING ALGORITHMS

One of the main methods to analyze microarray data is biclustering, a non-supervised technique widespread in the last years (see [84, 127] for biclustering surveys). Biclustering outperforms traditional clustering because of its two main characteristics: *simultaneous grouping of genes and conditions*, and *overlapping*. Simultaneous grouping means that *biclusters* (the groups found by biclustering algorithms) contain genes with similar behavior under a certain number of conditions (thus, the bicluster will group genes *and* the conditions under which the genes are related). Overlapping means that genes and conditions may be grouped together in more than one bicluster, so biclusters somehow can intersect (overlap) among them. Note that in the close technique of clustering, clusters group genes or conditions (but not both); and clusters rarely overlap.

On this chapter we set up the definition for biclustering and then, we review the search methods used by biclustering algorithms to find relevant groups, and the kind of groups they search for. Afterwards, we enumerate several tools to perform biclustering analysis. Finally, the major methods utilized to validate and compare biclustering results are described.

6.1 DEFINITION

Biclustering is a non-supervised classification method that, given a data matrix $A = a_{ij}$, groups rows with similar behavior under a subset of columns. From now on, we will consider gene expression data matrices, where rows are genes and columns are experimental conditions[1].

What is considered as similar behavior depends on the kind of biclusters that the method searches for, but typically it means that all the genes in the bicluster have expression levels within the same range or that the expression varies in the same fashion along the conditions. For example, the bicluster wrapped by a blue rectangle at the right bottom of fig. 21 has high expression levels for its genes under the first conditions of the bicluster but then the expression goes down for the last conditions. section 6.2 below cover the main kinds of bicluster.

Therefore, a bicluster $B = (G, C)$ is defined by the subset of genes G and the subset of conditions C that it groups together. They define a submatrix of A that contains the expression levels of $B = b_{ij}$.

Typically, a biclustering algorithm finds several biclusters. An important characteristic related to this is that these biclusters can *overlap*: they can coincide in one or more genes and/or conditions. We define the *overlap submatrix* as $O(B_1, B_2) = A(G_1 \cap G_2, C_1 \cap C_2)$. Note that $O(B_1, B_2)$ can have zero rows or columns, but not both. For example, the two biclusters at the right of fig. 21 overlap in several conditions, but not in genes.

1 We can define the *gene profile* of gene g_i as the expression levels of the gene along all the conditions of the matrix. Analogously, the *condition profile* of condition c_j are the expression levels of every gene for that condition.

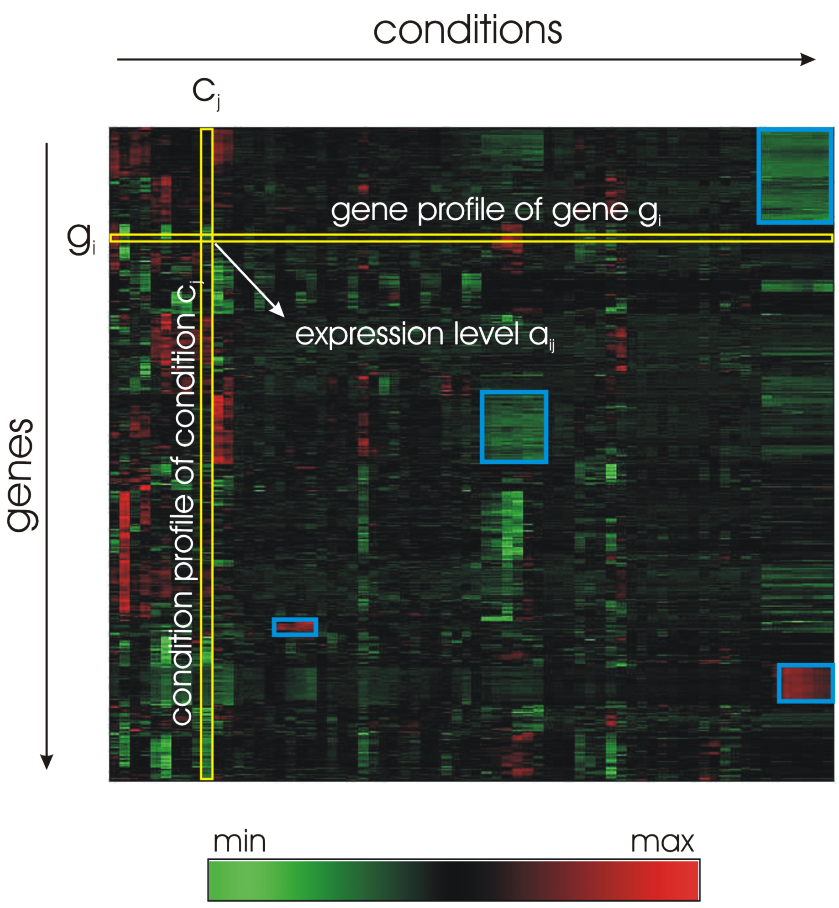


Figure 21: An example of gene expression matrix visualized as a heatmap, a representation where transcription levels are represented by a color scale (red conveys high expression and green conveys low expression, see chapter 7). Surrounded in blue, some possible biclusters. In yellow, two profiles, for gene g_i and condition c_j.

6.2 BICLUSTER TYPES

Biclustering, like clustering, relies on the concept of "similar behavior among individuals" (in this case, genes or conditions). Depending on how we define similar behavior, we have three main classes of bicluster [84] (see figs. 22, 23):

- *Constant value bicluster*: all the expression levels in the bicluster have exactly the same value (μ). These "ideal" bicluster condition is usually relaxed to a merit function with an interval $\mu \pm \delta$.
- *Coherent value bicluster*: the expression levels vary along rows and/or columns with some type of coherence, despite their overall level. This relationship may be additive or multiplicative, so rows and/or columns in the biclusters differ one to another in an additive or multiplicative factor (eqs. 6.1 and 6.2, respectively).

$$b_{ij} = \mu + \alpha_i + \beta_j \tag{6.1}$$

$$b_{ij} = \mu \times \alpha_i \times \beta_j \tag{6.2}$$

μ is the average background level for the whole bicluster, α_i is the factor for gene i and β_j is the factor for condition j.

Some authors call these factors shifting and scaling factors, respectively [3]. For example, the green bicluster in fig. 22 is an additive coherent bicluster, where we can consider $\mu = 0$, $\alpha_2 = 0$, $\alpha_3 = 1$, $\alpha_4 = 0$, $\beta_1 = 2$ and $\beta_2 = 4$. In the same figure, the red bicluster is a multiplicative coherent bicluster with $\mu = 2$, $\alpha_4 = 1$, $\alpha_5 = 1.5$, $\beta_1 = 1$, $\beta_2 = 2$ and $\beta_3 = 4$.

- *Coherent evolution bicluster*: in this case we just search for a qualitative rule of change in tendency but there is no quantitative restriction to it. For example, the orange bicluster in fig. 22 is a coherent evolution bicluster because the transcription levels of genes 5 and 6 increase from condition 5 to condition 6, but not in an additive or multiplicative way.

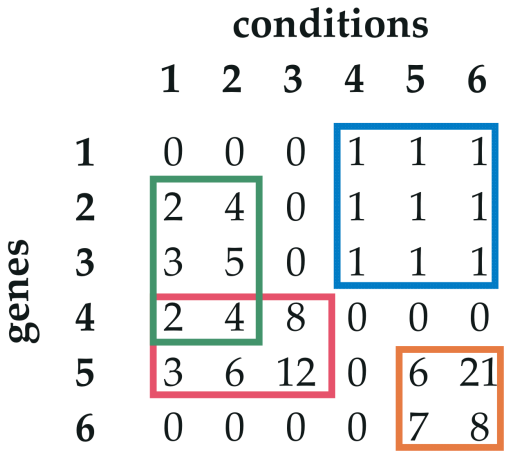
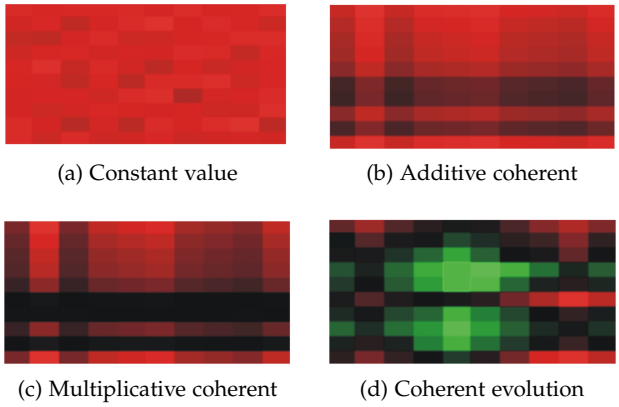


Figure 22: A simplified expression matrix with four possible biclusters. The blue bicluster is a constant bicluster. The green one is an additive coherent bicluster. The red one is a multiplicative coherent bicluster. The orange one is a coherent evolution bicluster by columns only.

Usually, the constraints defined by these types of biclusters apply to both row and columns. However, it is possible to apply them only to rows or to conditions. In the case of eqs. 6.1 and 6.2 we can achieve this by making α_i or β_j zero. The resulting biclusters are called *constant or coherent biclusters by rows or columns*. For example, the orange bicluster in fig. 22 has no coherence between conditions 5 and 6 (the first one increases from gene 5 to 6, while the second one decreases), so the biclusters has coherent evolution by rows only.

The constant model is strict but simple and is implemented, for example, by the Bimax algorithm [99]. The additive and multiplicative models are rich enough to model regulation processes without losing specificity. Therefore, most of the biclustering algorithms use variations of eqs. 6.1 and 6.2, such as [34, 79, 135, 26, 127, 77, 29]. Coherent evolution search has also been applied with success in some popular biclustering algorithms such as xMotifs [87], Statistic-Algorithmic Method for Bicluster Analysis (SAMBA) [126] and Order-Preserving SubMatrix (OPSM) [13].



(a) Constant value (b) Additive coherent

(c) Multiplicative coherent (d) Coherent evolution

Figure 23: Different types of bicluster as would be represented on a heatmap, where color conveys expression levels.

6.3 BICLUSTER SEARCH METHODS

After defining *what* biclustering type we search for, we have to define *how* we do it. Again, there are several options, but it is not as critical for the heterogeneity of biclustering results as the type of bicluster. Following the classification of Madeira and Oliveira [84] we have:

- *Iterative row/column clustering*: the straightforward approach, applies clustering algorithms to the rows and columns of the expression matrix and then combine them to build biclusters. It is also called two-way clustering [53, 128] or conjugated clustering [26].
- *Divide and conquer*: this method breaks the problem into several subproblems (which often means several submatrices) to be solved and then combine the partial results to generate the solution. Hartigan's pioneer biclustering method [56] can be classified as a divide and conquer algorithm. See also [40, 131].

The first biclustering method appeared in 1972 [56], but its first application to microarray data was in 2000 [34]

- *Greedy iterative search*: this method assumes that to find an optimal local solution can lead to a globally good solution. Cheng and Church biclustering algorithm (CandC) [34], FLOC [145, 146], OPSM [13], xMotifs [87] and δ -patterns [27] are all examples of greedy iterative searches.
- *Exhaustive bicluster enumeration*: searches for all the possible biclusters following some criterion [127, 126, 80, 90]. This is time consuming and usually they are combined with restrictions on the size or number of biclusters, or by pre or post-filtering of results [99].
- *Distribution parameter identification*: assumes that the data structure follows a statistical model, trying to fit its parameters to the data by minimizing a certain criterion through an iterative approach. Plaid models [79, 135], Spectral biclustering [77] and Rich Probabilistic Model (RPM) [112, 111] are some examples of this kind of biclustering.

6.4 BICLUSTERING TOOLS

Some of the biclustering algorithms described above are published with available implementations, sometimes in the form tools. Also, some authors have compiled several biclustering implementations for comparison with their own methods or to make biclustering a more available tool for non-experts in the field. Here we enumerate the most interesting ones:

- Biclustering Analysis Toolbox (BicAT) [9]: possibly the most comprehensive tool for biclustering, BicAT supports five biclustering algorithms (Bimax, CandC, OPSM, SAMBA and xMotifs), along with the two main clustering methods (hierarchical and k-means). This software permits to do some pre-processing of data (normalization and binarization) and post-processing (search, filtering and gene pair analysis). It also lists the bicluster results and visualizes them by means of a heatmap and parallel coordinates (see chapter 7 for a wider discussion about bicluster visualization).

- EXPresion ANalyzer and DisplayER (Expander) [115] is another interesting tool about biclustering. Although it supports just a biclustering algorithm (SAMBA [126]), it also implement several clustering analysis (CLICK algorithm [117], k-means and SOM), pre-processing (filtering and normalization) and visualizations (heatmaps, parallel coordinates -called contours-, dendrograms, etc.). It also allows to load your own clusters or biclusters and inspect them with their visualizations.

- Gene Expression Mining Server (GEMS) [144] is an online tool to perform biclustering analysis by Gibbs Sampling (similarly to [119]). Although it only provides one biclustering method, it is, to our knowledge, the only example of online biclustering tool.

- BiVisu [33]: is a simple tool for biclustering analysis and visualization. It offers a search method for additive or multiplicative coherent biclusters, and visualizes the results with parallel coordinates. It also implements some pre-processing and post-processing utilities.

- HCE [113, 114], although for generic use and just for traditional clustering, it is an example of exhaustiveness on statistical analysis and of visual support and human-computer interaction. It offers pre-processing by means of filtering and transformation, hierarchical and k-means clustering, and several post-processing methods. About visualization, dendrogram+heatmap, histogram, parallel coordinates and scatterplot are available with a high degree of interaction.

Apart from these biclustering tools, there are several other clustering and gene expression analysis tools, such as gCLUTO [101] which provides a novel visualization (mountain maps) and new ways to interact with the dendrogram+heatmap visualization (see section 7.2), or ExpressionProfiler [74], a collaborative web-based platform for microarray gene expression, very complete, specially regarding the pre-processing and clustering analysis.

Some of these tools also integrate biological knowledge, such as GO terms. Table 3 summarizes the characteristics of the considered tools.

Note how several tools focus on just a biclustering algorithm, and some of them also offer clustering as an alternative. Following the features of these tools, one can conclude that there is consensus in the need for the pre-processing steps of normalization, filtering and transformation, and for post-filtering. Also, there is a lack of integration of biological knowledge within the tools.

TOOL	Biclustering	Clustering	Pre-processing	Post-processing	Biological knowledge
BicAT [9]	5 methods	2 methods	normalize, binarize	filter	none
Expander [115]	1 method	4 methods	normalization, filter	filter, PCA	none
GEMS [144]	1 method	none	normalize, filter, transform	none	none
BiVisu [33]	1 method	none	transform	filter	none
HCE [114]	none	2 methods	normalize, filter	compare 2-by-2	Affy annotation
gCLUTO [101]	none	3 methods	none	none	none
ExpressionProfiler [74]	none	2 methods	normalize, filter, transform, impute	compare	GO, ChroCoLoc

Table 3: Summary of the main characteristics of biclustering tools.

6.5 VALIDATION OF BICLUSTERING ALGORITHMS

Biclustering validation is partially based on clustering validation, which consists on the calculation of validation indices. There are three main indices for cluster validation ([70], chapter 4):

- *External indices*: they measure the precision by which the clusters match with embedded structures that we know that exist in the data. Two and single-matrix validation measures can be defined. In single-matrix indices, the measure is computed from a matrix that has as rows the found clusters and as columns the groups known to be in the matrix. The index reveal the percentage of matches between the real groups and the found clusters, in terms of *sensitivity* (the percentage of real groups covered by the clusters) and *specificity* (the percentage of clusters that are related to real groups). In the case of the two-matrix technique, two binary matrices are built, one for the clusters in the results and another for the a priori groups. Indices are computed from these matrices, such as the Rand index, the Jaccard coefficient, the Hubert Γ statistic, the Minkowski measure or the Folkes and Mallows measure [55].

specificity is also called relevance, *and sensitivity is also called* module recovery

- *Internal indices*: they compare the intrinsic structure of data with the clusters found. No information apart from the raw data is needed. The Pearson's Cophenetic Correlation Coefficient (PCCC) [55], silhouette widths and the Dunn's validation index ([15], chapter 13) are typical internal indices in clustering validation.

- *Relative indices*: they compare the clustering algorithm with itself, by comparing results under different parameter configurations. It is a way of finding the best configuration of the algorithm for a given input. Relative indices are usually external or internal indices used for this goal.

External indices, in particular single matrix indices, have been adapted to biclustering in literature [134, 99]. The adaptation of internal indices to biclusters is more complex, and to our knowledge it hasn't been addressed in literature yet. Relative indices have been used to find stability when the algorithm has pseudo-random behavior [29], but not to search for optimal initial parameters.

On the other side, the goodness of biclustering results are also measured in terms of how well do they match with the previous biological knowledge about genes. We call this *biological validation*, and it can be classified as an external validation, since they rely on metrics calculated from a priori (biological) knowledge in order to measure the goodness of results.

GO terms and biological networks have been used for the computation of biological external indices [99, 29, 90]. In both cases, we search for statistical *enrichment* of genes: the bicluster groups several genes with the same GO terms or network features[2] that wouldn't be grouped by chance. The statistical enrichment is usually calculated by means of a *statistical significance test* that calculates the *p-values* for each biological

2 For example the network motifs, simple structural blocks, such as fast forward loops or bi-fans [85]

feature. This p-value is the probability that the biological feature has been grouped by chance. Then, a significance level α is set, and only the features with a p-value lower than it are selected. The features (if any) under the significance level are said to be significantly enriched by the group of genes.

There is controversy among statisticians about the appropriate use of statistical significance tests [7, 63], but it is frequently used in biclustering validation and comparison (see section 6.6).

6.6 COMPARISON OF BICLUSTERING ALGORITHMS

We identified three major biclustering comparisons, those of Prelic et al. [99] (P), Reiss et al. [103] (R) and Okada et al. [90] (O). These comparisons cover broadly the comparison process and take into account a large number of biclustering algorithms. All of these comparisons include Bimax, ISA, SAMBA, CandC and OPSM biclustering algorithms, adding some other biclustering and clustering algorithms on each case (see table 4).

Prelic et al. comparison is based on three validation methods:

- *External validation*: against synthetic matrices with embedded constant and additive coherent biclusters, with different degrees of overlap and noise. To measure how well the found biclusters match the embedded biclusters a *gene match score* is computed, similar to Turner et al. specificity and sensitivity measures [135][3], but just for genes.

- *GO enrichment validation*: a hypergeometric test for GO enrichment is performed. Briefly, this statistical significance test measures if the genes in the biclusters are annotated with GO terms that would hardly be grouped by chance (the term is *enriched* by the bicluster). If at least one MF or BP GO term is enriched by the bicluster, for a given significance level, the bicluster is considered as significant. The percentage of significant biclusters for each significance level is the measure of enrichment.

- *Pathway and PPI enrichment validation*: for each bicluster $B = (G, C)$, the number of disconnected genes among G in the corresponding network is calculated, and also the average path length among genes in G in the same network. Theoretically, both of them should be smaller in a bicluster than in a random group of genes. A statistical significance test (in this case, a Z-test) is performed, and the percentage of enriched biclusters is given, like in the previous case.

Note that none of these validations take into account conditions, because hierarchical clustering is one of the algorithms to be compared and it does not group conditions. All the validations use external indices, either specificity/sensitivity measures or significance tests.

Okada et al. use the same validations, except for the pathway enrichment validation. They maintain gene-based indices, such as the gene score match, although they do not include clustering algorithms in the comparison.

3 see section 11.2

Reiss et al. introduce some innovations: motifs enrichment and the division of bicluster results into two sets. *Motifs* are simple graph structures, such as two nodes connected to the same two other ones (a *bi-fan*) or a node *a* connected to nodes *b* and *c*, with node *b* also connected to *c* (a *fast-forward loop*). These structures occur frequently on regulatory networks, so the study of motif enrichment is an interesting approach. The second innovation, splitting biclusters into two result sets (one half containing the larger biclusters and the other half for the smaller ones), tries to avoid the bias on indices due to biclustering algorithms that find very large biclusters. Unfortunately, this problem is not completely solved by the split (for example, the OPSM algorithm outputs biclusters so large that to split them into two subsets do not remove the bias). Finally, Reiss et al. do not calculate gene match scores.

Algorithm	Gene match score				GO score				Total
	P	O	R	Avg	P	O	R	Avg	
ISA [14]	2	1	-	1	3	4	3	2	1
Bimax [99]	1	4	-	3	2	3	6	3	2
SAMBA [126]	3	3	-	4	4	5	2	3	3
CandC [34]	5	5	-	6	5	6	7	6	4
OPSM [13]	6	6	-	7	1	2	1	1	-
xMotifs [87]	7	6	-	8	6	7	-	7	5
hclust	4	-	-	5	6	-	7	7	4
k-means	-	-	-	-	-	-	4	4	-
BiModule [90]	-	2	-	2	-	-	-	-	-
cMonkey [103]	-	-	-	-	-	-	5	5	-

Table 4: Biclustering algorithms ranking. P, O and R refer to Prelic et al., Okada et al. and Reiss et al. comparisons, respectively. Ranks have been interpreted from published figures and tables. AVG is the average rank for each score, and Total is the average rank of AVG ranks. The Total score ignores algorithms with just one score, and OPSM (see text).

A biclustering algorithms's ranking[4] is proposed in table 4. Following, some considerations about the results are presented:

- Bimax, ISA and SAMBA are probably the best choices for biclustering, because they obtain high ranks in every comparison, from papers or works not carried out by their authors themselves. The good results of these algorithms with synthetic data are confirmed by biological validation. In particular, we selected Bimax for several visualization examples along this thesis because of it, and because it has an easy interpretation of biclusters (highly up or down regulated constant biclusters).

4 These ranks are qualitatively inferred from the figures depicting the results of the corresponding papers. In the case of several scenarios for a measure, like in the case of gene match score for overlapping modules and noise values, we calculated the average of the qualitative ranks. PPI and Metabolic Pathway enrichment ranks are not included because they vary too much and, in some cases, are ambiguous [99]

- CandC and xMotifs yield a lower performance. However, CandC is always present in comparisons because it is the first biclustering algorithm applied to bioinformatics.
- The results of clustering algorithms are not as bad as one could expect. This is partially because gene-centered measures are being used, ignoring conditions.
- BiModule and cMonkey are only compared in one paper, carried out by their respective authors, so the results about them are not conclusive. In particular, the design of cMonkey is very biological data-dependent, which could explain the lack of gene match scores with synthetic data for its validation and comparison.
- OPSM validation presents several problems. Regarding synthetic data validation, this algorithm searches for a very broad definition of coherent evolution biclusters (preservation of order ranks), which gives way to very large biclusters. In terms of gene match scores, although OPSM should be capable of finding the embedded biclusters (specially coherent biclusters), it adds several irrelevant rows and columns. Regarding biological data validation, it is highly probable that a very large bicluster would enrich at least one GO term. In part, this is a problem of the significance tests with large groups of genes, identified by [103].

6.7 CLUSTERING AND BICLUSTERING

Biclustering and clustering present fundamental differences in terms of elements grouped (in biclustering, both genes and samples) and in terms of overlapping (usually not available in traditional clustering). Both methods share the search for groups internally coherent, but biclustering does not normally require inter-cluster separation because of the possibility of overlap. Cluster and bicluster validations have also different implementations due to these two characteristics of bi-dimensionality and overlapping.

Despite these differences, some authors classify biclustering as a kind of clustering, and biclustering publications usually compare their results against clustering algorithms. Also, some biclustering tools implement clustering methods. The fact that both techniques share several concepts is unavoidable: the search for groups of elements with similar behavior, the non-supervised approach to classification and the searches based on similarity metrics are some examples of it.

Therefore, are their differences greater than their similarities or vice-versa? It depends on the application field and on each particular dataset. Regarding gene expression analysis, the special characteristics of biclustering imply an improvement for the modeling of expression profiles. However, for gene expression matrices with very low number of conditions or well known and controlled conditions, the advantage can be spurious. In practice, clustering is still largely used for gene expression analysis. Microarray data are so rich on information that simple clustering usually reveals classifications which satisfy the main questions of biologists. However, several secondary questions, such as subtler interactions among genes, or relationships among genes under different experimental conditions (which are still out of the scope of most publications) could be more easily answered by means of biclustering.

7

VISUALIZATION IN GENE EXPRESSION AND BICLUSTERING

Gene expression matrices, because of the high dimensionality and, consequently, the difficulty to detect their inherent structure, prove to be a fertile field to design visualizations. The complexity of some analysis methods, such as clustering hierarchies or bicluster overlapping adds more challenging tasks to their visualization.

On this chapter we review the main solutions adopted in literature in order to visualize gene expression matrices and the results from clustering and biclustering algorithms. Finally, we summarize the most relevant visualization tools for microarray data analysis.

7.1 VISUALIZATION TECHNIQUES FOR GENE EXPRESSION MATRICES

The entities involved in a gene expression matrix are easy to identify: genes, conditions and expression values. The graphic elements used to convey these entities define the resulting visualizations. Not less important than the entities is the knowledge we want to discover in the expression matrix. This can be summarized in two questions[1]:

- *Is there structure within the microarray?* or, more precisely, *are there groups of features related?*
- *What is that structure?* or *how do these groups relate?*

In order to represent such entities and answer such questions, two main visualization techniques have been successfully applied in bioinformatics: heatmaps and parallel coordinates (see table 5).

ENTITY	HEATMAP	PARALLEL COORDINATES
Gene	y-axis	polyline
Condition	x-axis	x-axis
Expression level	color	y-axis
Structure	reordering	filtering

Table 5: Summary of the encoding of gene expression data entities in heatmaps and parallel coordinates

1 Note that the entities and questions that we consider for information visualization are similar to the ones considered in microarray data analysis (chapter 6), which will serve as an introduction to the concepts of visual analytics in the next chapter

7.1.1 *Heatmaps*

A heatmap is the natural visualization of microarray data, due to the aspect of a microarray after fluorescent stimulation (see figs. 5 and 21). A heatmap is a 2D representation that uses cartesian axes to display the dimensions of the data matrix. Usually, genes are displayed along the y-axis and conditions along the x-axis. Each expression level a_{ij} is drawn as a rectangular shape at the corresponding (x, y) location, and it is colored according to a color scale that often varies from a bright color c_d for the lowest expression value to an intermediate color c_0 to another bright color c_u for the highest expression value. Traditionally, c_d is green, c_0 is black and c_u is red, in order to match with the typical fluorescent dyes in DNA microarrays.

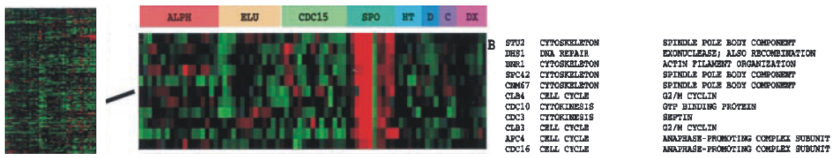


Figure 24: The sporulation group of genes in *Saccharomyces cerevisiae* discovered in the work of Eisen et al. [44]. The visualization already used a labeling technique by biological terms for genes and color labels for groups of conditions (reproduced from [44]).

A static heatmap is not very useful to convey structure information unless rows and/or columns are reordered following some criterion. Eisen et al. [44] were the first ones to display gene expression data from microarrays as heatmaps, reordering rows to convey groups found by means of hierarchical clustering (see fig. 24). This work also displayed the hierarchical clustering dendrograms along with the heatmap (see section 7.2), and set the grounds for the biological validation of groups. This visualization scheme is still extensively used after more than ten years [5, 136, 31, 29, 90].

Unfortunately, heatmaps show some drawbacks, specially the fact that the heatmap by itself is not good to reveal structure, it is hard to explore and it is dimensionally unbalanced: gene expression matrices are much taller (around $10^{[3-4]}$ rows) than wide (around $10^{[1-2]}$ columns). In addition, the green-black-red color scale for expression levels is not the best choice (human perception ability to distinguish color scales depends on the hue, see [141] pages 129–132), and on information visualization grounds, a blue-white-red or just a grey scale are usually accepted as better. Some authors opted for blue-black-yellow [115] or green-white-red scales [101]. *Gehlenborg et al.* [51] add up a relevance color scale on blue hues that overlaps to the expression level scale in order to represent their associated p-values (see fig. 25).

Possibly the best approach to color scales is the one of *Hibbs et al.* [60] who use a grey color scale and perform a previous ranking of expression levels in order to be more robust to noise. The grey hues are known to be better perceived than other hues [141], while the ranking avoids misinterpretations due to color scales (see fig. 26).

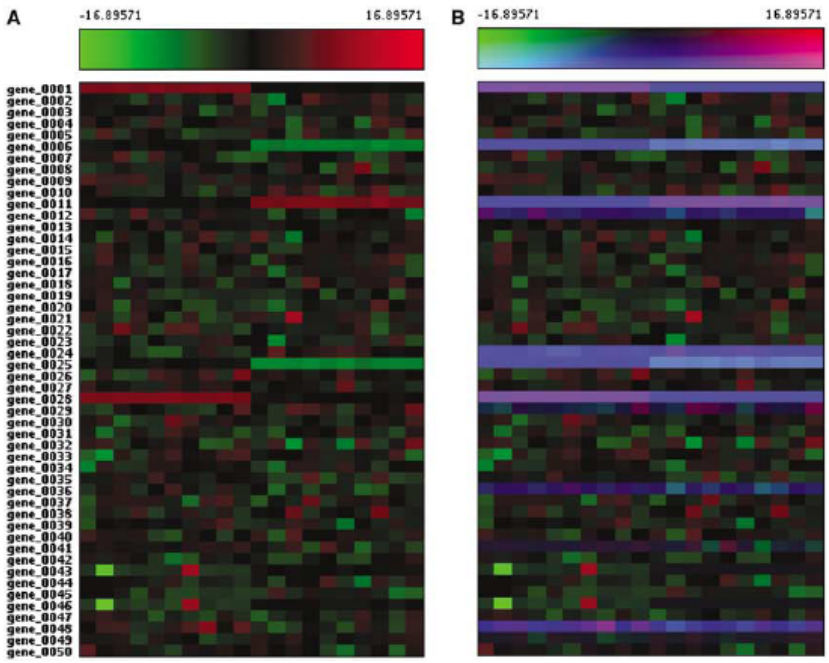


Figure 25: The typical heatmap visualization (left) is enhanced by a blue gradient conveying the relevance of expression levels following a statistical test (right). Reproduced from [51].

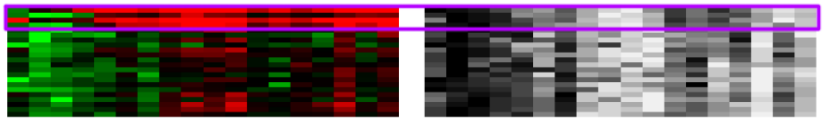


Figure 26: The usual heatmap visualization of a cluster (left) shows four genes (within the purple box) that look be very different from the rest. Oh the right, the expression level ranking and the change of color scale reveal that they follow the same general pattern of the rest of genes in the cluster. Reproduced from [60].

Apart from color scales, microarray heatmap improvements focused on the interaction, in order to reduce the complexity to explore and minimize the dimensional unbalance. The typical solution for this is to apply zooming techniques [101, 115, 99]. *Treeview* [104] implements a focus+context approach with a visualization representing the whole microarray and another one for smaller subsets selected by drawing a rectangle. In Treeview, additional visualizations represent array names and gene annotations and, if hierarchical clustering is performed, the corresponding dendrogram (see section 9.2 for more details). *gCLUTO* [101] implements a collapsing branch technique that depends on the use of hierarchical clustering (see section 7.2).

7.1.2 *Parallel Coordinates*

Parallel coordinates [67] have also been used to visualize microarray data, specially subsets of genes. In this technique, each gene profile g_i is considered as a m-dimensional point $p_i = (a_{i1}, a_{i2}, ...a_{im})$ where a_{ik} is the transcript abundance of g_i under condition c_k. Conditions are represented as vertical lines along the x-axis at equidistant points $x^c_1, ..., x^c_m$. Each gene profile g_i is displayed as a polyline of m points (x^c_k, y_k), with y_k proportional to a_ik (fig. 27).

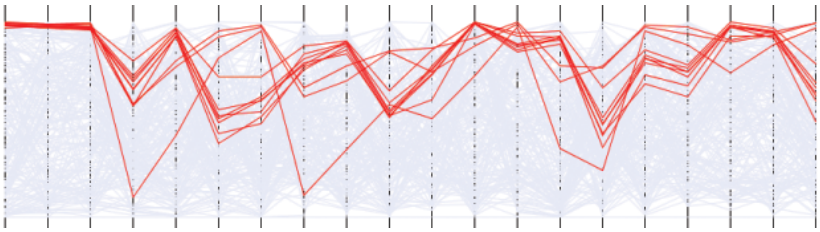
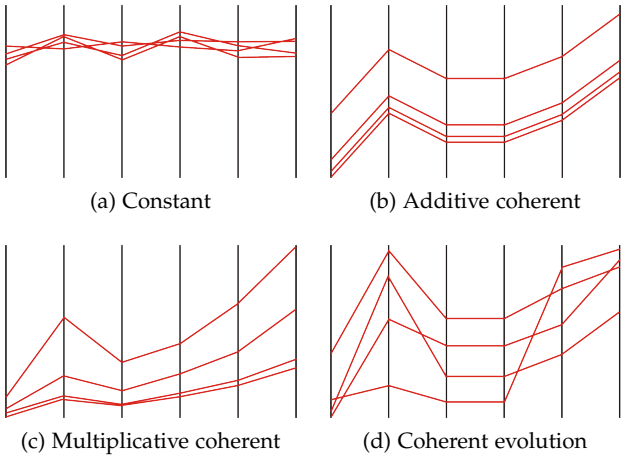


Figure 27: Parallel coordinates visualization of a 200×20 synthetic gene expression matrix. It is clear that to display every polyline, even in the background with a neutral color, can clutter the visualization. It is more usual to display just a group of genes (in red).

This technique solves the dimensional unbalance of heatmaps by assigning the y-axis to transcription levels instead of to individual genes, but at the cost of cluttered profile polylines. Therefore, parallel coordinates cannot be applied *as is* to visualize whole expression matrix, but to visualize selected groups (clusters) of genes [34]. Besides, human perception is good at interpreting line patterns such as parallel lines, mirror effects and changes in slope; outperforming color scales [141]. Regarding biclusters, this is specially useful to distinguish among bicluster types, specially in the case of constant and coherent (additive or multiplicative) evolution (see fig. 28).



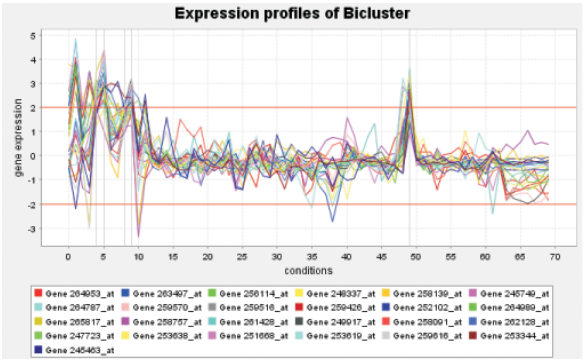
(a) Constant (b) Additive coherent

(c) Multiplicative coherent (d) Coherent evolution

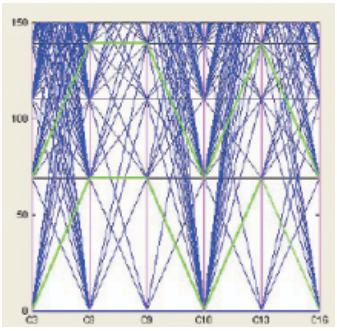
Figure 28: Parallel coordinates representation of different bicluster types.

HCE [114] has one of the best and more interactive implementations of parallel coordinates. It simplifies the representation of large sets of polylines by using a polygon to avoid cluttering while keeping the context, and implements dimensional, model and text brushing in order to select reduced groups of lines, thus offering a kind of visual clustering. HCE is further described on section 4.2.6.

In order to display a bicluster with k genes and s conditions we can draw the k polylines corresponding to the genes, and rearrange together or highlight the axes corresponding to the s conditions. BicAT [9] and BiVisu [33] use parallel coordinates to display single biclusters (see fig. 29). However, their representations are limited. BicAT does not rearrange bicluster conditions, it simply marks their corresponding axes with vertical lines (making hard to visualize the whole bicluster). On the other hand, BiVisu only visualizes the segment of the polyline corresponding to the conditions in the bicluster, losing context information for other conditions. None of both methods provide interactive thresholds to manipulate the display.



(a)



(b)

Figure 29: a) BicAT parallel coordinates for a 21×5 bicluster (generated with [9]). b) BiVisu parallel coordinates (reproduced from [33]).

7.2 VISUALIZATION TECHNIQUES FOR CLUSTERING AND BICLUSTERING

We have seen in the previous sections that the visualization of data and the visualization of groups within the data are highly related, usually by means of the reorganization of the visualization according to groups. This is a good approach for clustering, where each element in the data is in one (and just one) cluster.

In order to visualize the whole results of clustering, the most used technique is the dendrogram+heatmap approach (see fig. 30). In this technique, a dendrogram represents the decision tree of hierarchical clustering, usually coloring the groups formed at the threshold cut. The heatmap is represented sideways, reordered to fit the clustering.

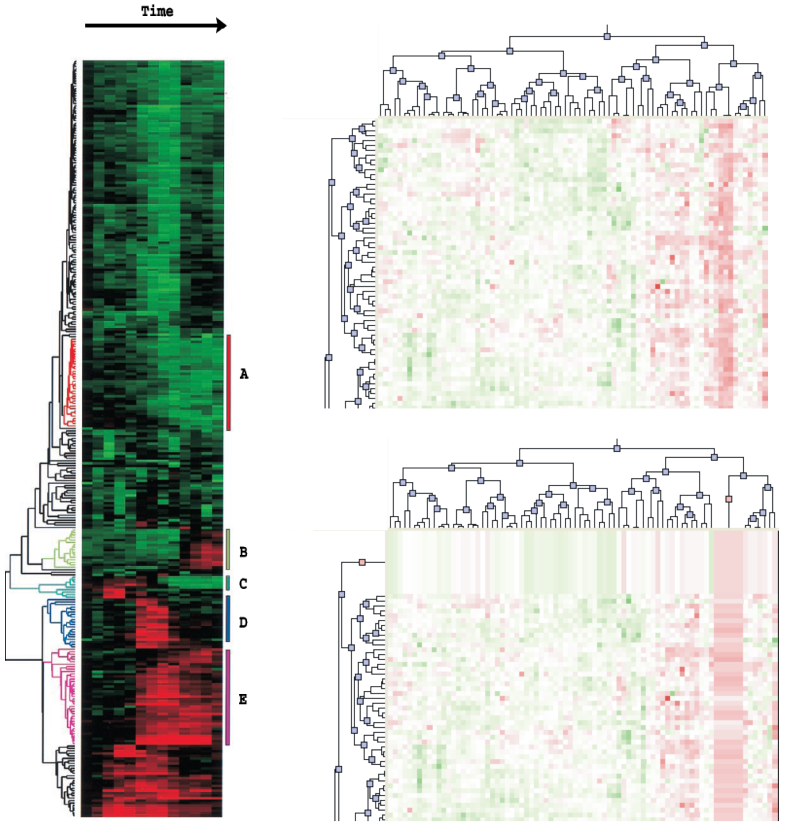


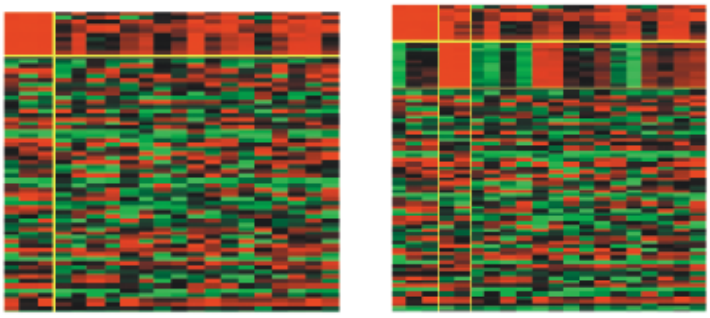
Figure 30: a) The dendrogram corresponds to a hierarchical clustering and the rows of the heatmap are reordered accordingly, revealing gene expression patterns. Reproduced from [44]. b) Detail of a "two dendrograms+heatmap" visualization (top). Branches can be collapsed to facilitate the navigation (bottom). Generated with gCLUTO [101].

Note that this two-way hierarchical clustering is a concept very close to biclustering.

The obvious step to follow in clustering visualization is to add the clustering of columns in the same way than rows, resulting in "two dendrograms+heatmap" visualizations, implemented for example in Treeview [104]. HCE [114] also implements this approach, and allows to modify the clustering threshold in order to define the clusters (see fig. 9). gCLUTO adds the filtering of groups by collapsing branches and

drawing a rectangular area whose color is the average of the expression levels included in the rectangle, and whose size depends on the number of genes and condition under the collapsed branches (see 30b).

The adaptation of heatmaps to represent a single bicluster can be easily done by reordering the rows and columns of the heatmap so the genes and conditions in the bicluster will be together in, say, the top-left corner of the matrix (see fig. 31a). BicAT [9] implements a good example of it.



(a) (b)

Figure 31: a) A bicluster at the top-left corner of the expression matrix, delimited by yellow lines. b) Additional biclusters can be visualized by reordering rows and columns through the diagonal of the matrix, providing that they only overlap with the previous and following bicluster (figures generated with *biclust* [72])

Unfortunately, a heatmap presents geometrical limitations to visualize several biclusters simultaneously, specially if they overlap (see fig. 31b). We can represent n sorted biclusters in a heatmap only if each one has just rows or columns in common with its previous and following biclusters, and not with any other bicluster. BiVoc [54] addresses this problem by repeating rows and columns to properly represent overlapped biclusters (see fig. 32).

Another technique not bound to the representation of expression levels is the mountain map [101]. In this visualization an independent entity (a *mountain*) is assigned to each cluster, mapping display characteristics to the cluster characteristics. As in parallel coordinates, each gene profile is considered as a m-dimensional point, and a cluster is defined by the m-dimensional midpoint, computed as the average of all the genes it contains[2]. MultiDimensional Scaling (MDS) [78, 11], a type of PCA, is the used to map the m-dimensional midpoints to 2D points. These points are the locations of mountains in the visualization. Other characteristics of the mountain (height, volume and color) are defined by cluster parameters (internal similarity, number of genes, internal deviation). The result is a 3D colored mountain-like terrain which reveals details hard to see in the dendrogram+heatmap visualization (see fig. 33).

2 We will consider the application to gene clustering, it is analogous for condition clustering.

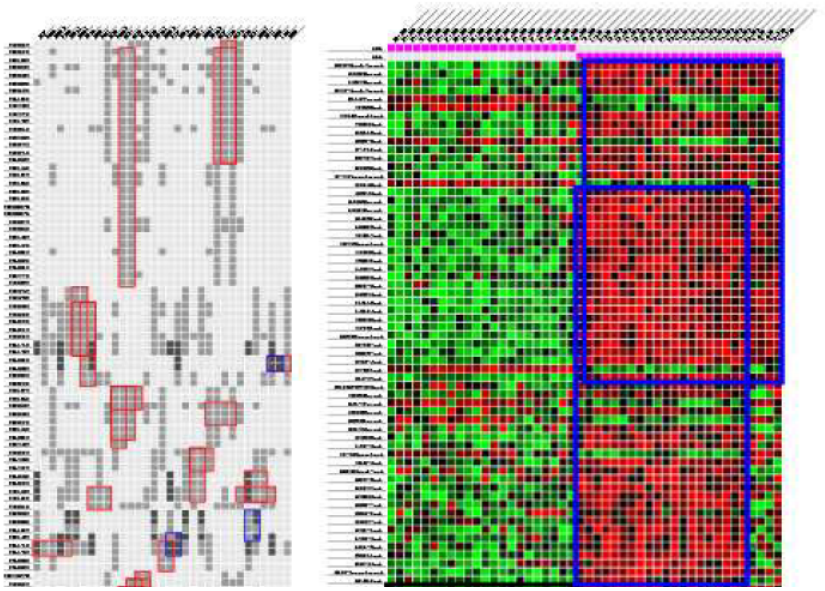
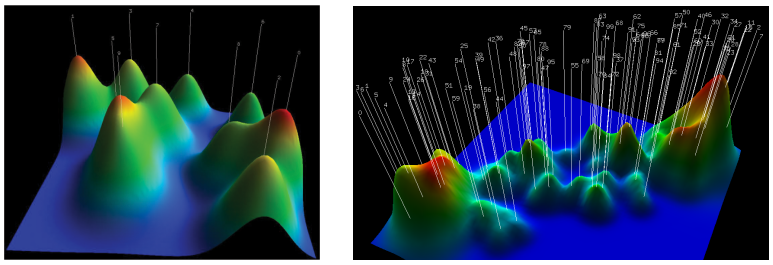


Figure 32: Detail of BiVoc [54] visualization of biclusters (within red and blue rectangles) on two different expression matrices. In order to be able to visualize several biclusters, rows and columns may be replicated in the heatmap. Reproduced from [54].



(a) (b)

Figure 33: a) Mountain map of 10 clusters. Three of them have low internal deviation (red summit). b) The mountain map of 100 clusters reveals two groups of several clusters, we call them "superclusters", at the sides (figures generated with [101]).

7.3 MICROARRAY VISUALIZATION TOOLS

On the previous sections of this chapter we have cited several tools that implement visualization techniques related to microarrays. Most of them are also clustering or biclustering analysis tools (see table 3), so they approach to some extent to the visual analytics point of view, integrating analysis and visualization of results. Table 6 summarizes the main characteristics of these tools. Note that this is not a comprehensive list of tools for microarray data analysis, and it is focused on tools that implement biclustering or other interesting group visualizations such as gCLUTO. There are many other commercial tools focused on microarray data and traditional clustering similar to HCE, specially SpotFire®[3] and GeneSpring®[4]. See [88] for a comparative analysis of these other tools.

3 SpotFire®, Decisionsite™ for functional Genomics, www.spotfire.com
4 GeneSpring®, cutting edge tools for expression analysis, www.silicongenetics.com

TOOL	Heatmap (HM)	Parallel Coordinates (PC)	Other visualizations	Degree of interactivity
HCE [114]	yes (with dendrogram)	yes	scatterplot, histogram	high
gCLUTO [101]	yes (with dendrogram)	no	mountain map	medium
Treeview [104]	yes (with dendrogram)	no	zoom and annotation views	medium
ExpressionProfiler [74]	yes (with dendrogram)	no	density plot	low
BicAT [9]	yes (one bicluster)	yes	none	low
Expander [115]	yes (one bicluster)	yes (called patterns)	PCA scatterplot	low
BiVoc [54]	yes (several biclusters)	no	none	none
BiVisu [33]	no	yes	none	none

Table 6: Summary of the main visualization characteristics of the surveyed tools. The tools above the horizontal line deal visualize clustering results, the tools below line deal with biclustering results.

8 VISUALIZATION OF DATA GROUPS

Several data analyses end up inferring relationships that can, in the end, be considered as groups. Clustering searches for clusters, biclustering for biclusters, supervised classifiers for classes, etc. In addition, several datasets contain inherent groups. For example, a database of scientific papers can be seen as a database of scientists, classified by groups of co-authorship.

The visualization of groups has been addressed in several forms, frequently by using some kind of set diagram or graph. The following sections review some of the most relevant approaches to the visualization of groups.

8.1 SET DIAGRAMS

Euler diagrams are the main method to visualize groups, where each set is represented by a *contour* (a closed curve). The area described by any intersection, union or difference between two or more contours is a *region*. Recursively, any area described by the intersection, union or difference between two or more regions is also a region. Contours can also be considered regions themselves, sometimes called *basic regions* [62]. Finally, a *zone* is a region that does not contain any other region (also known as minimal regions). See fig. 34a for an example of contours, regions and zones.

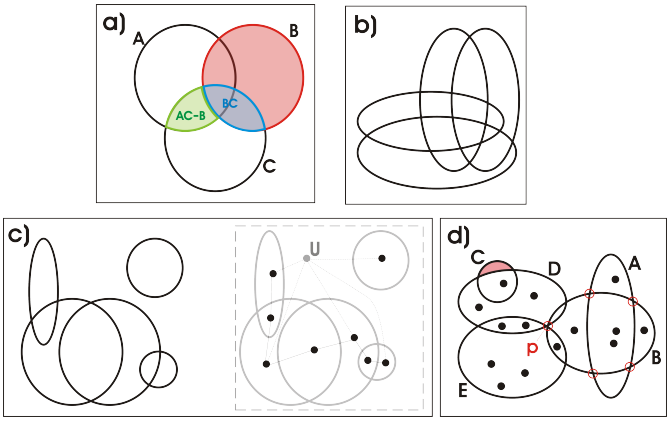


Figure 34: a) Venn diagram. B is a contour, $B \cap C$ is a region, and $A \cap C - B$ is a zone b) Venn diagram for four sets. c) Euler diagram (left) and its corresponding dual graph (right). d) Hypergraph drawn in the subset standard. Empty zones (in red) and any contour crossings (red circles like p) are allowed.

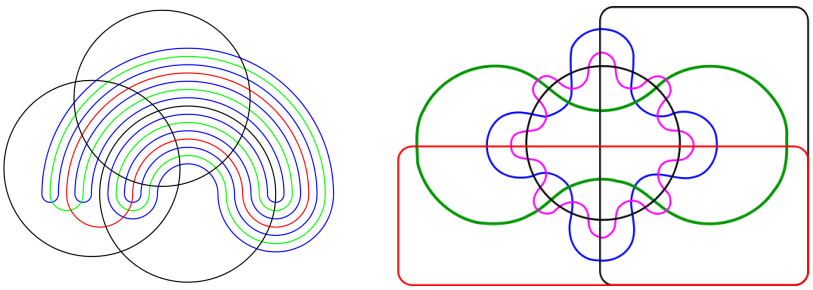
Euler diagrams were named after the work of Leonhard Euler in the 18th century. The same happened with John Venn and the Venn diagrams around 1880

A *Venn diagram* is an Euler diagram in which all intersections among contours must occur (fig. 34b). Venn and Euler diagrams are abstract diagrams, they do not define rules about the elements within the sets or how they are drawn.

Graph-enhanced Euler diagrams define a graph, an underlying Euler diagram and a mapping from the graph nodes to the zones of the Euler diagram. This is usually done with the help of a *dual graph*: a graph that assigns one node to each zone and joins adjacent zones by edges (the empty space in the bounding rectangle is sometimes identified as another zone, U, see fig. 34c). Region topology is very restricted in Euler diagrams, for example, empty zones[1] are not permitted, and the intersection of contours must be by means of just two intersection points.

On the other side, *hypergraphs* are graphs in which edges (hyperedges) join one or more nodes, instead of just two nodes. From our point of view, each hyperedge can be considered as a group. Hypergraphs can be drawn following two main standards [17]: to draw each hyperedge by connecting the points that represent their vertices (*edge standard*), or to represent each hyperedge by a closed curve that contains these points (*subset standard*)[2]. Hypergraphs drawn in the subset standard are similar to Euler diagrams, but do not take into account regional constraints, focusing in nodes inside the sets more than in the containers. For example, the hypergraph drawn in fig. 34d has empty zones (as the one marked in C), contour crossings with more than two points (the contours of A and B, for example, intersect in four points) or intersection points of more than two contours such as p. All these circumstances are not permitted in Euler diagrams.

From an information visualization point of view, the success of Euler diagrams to convey relationships among groups relies on the Gestalt principles of closure and continuity (see section 4.2.4). Ware [141] states that the use of texture and color can convey a more complex set of relations than the use of just closed contours. Anyway, the immediacy of perception for a low number of sets is quickly lost when this number grows (see fig. 35).



(a) Venn diagram for six groups (b) Edward's Venn diagram for six groups

Figure 35: Venn diagrams for 6 groups. The representation becomes cluttered because of an excess of parallel contours. Even the use of symmetry in the Edward's solution does not clarify the diagram.

1 Zones without nodes inside.
2 There are some other ways to draw hypergraphs, see [75]

Euler diagram drawing for up to three sets has been addressed [46]. In addition, different aesthetic metrics are applied [47] to make the diagram more readable. With a more flexible, extended definition of Euler diagrams, up to eight sets can be represented without zone errors [139]. In this case, a contour segment can belong to more than one set and zones may not be convex and can have holes. However, the use of non convex contours and holes may not be as intuitive as simple closed curves. On the other hand, hypergraphs have a less formal definition and any number of subsets can be drawn. Bertault and Eades [17] propose several methods to build the graph corresponding to a given hypergraph, with good results for small hypergraphs, but it becomes too cluttered when the number of nodes, the size of hyperedges and the degree of overlapping grow. 20 elements, with about 10 hyperedges of length (at most) 5^3 are enough to clutter the representation. Omote and Sugiyama [92] propose a method with an exhaustive set of rules in order to visualize groups avoiding cluttering.

Hypergraph drawing is the only method with the capability to show several overlapped groups as contours, while keeping the visualization of all elements and group relationships within a single diagram. From an information visualization point of view, it is necessary to design proper visual elements and implement interaction in order to minimize the cluttering and other scaling issues that may arise.

8.2 CLUSTERED GRAPHS

Clustered Graphs represent non-overlapped groups, either inherent to the data or obtained by clustering techniques, usually hierarchical clustering (see fig. 36).

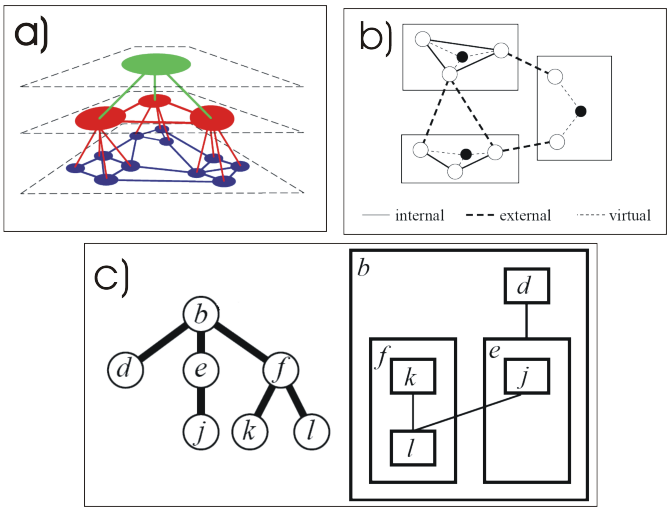


Figure 36: a) Hierarchical clustered graph. b) Force directed clustered graph. c) Compound graph.

A *Hierarchical Clustered Graph (HCG)* [42] starts with the drawing of the highest level of a hierarchical clustering (only one cluster including

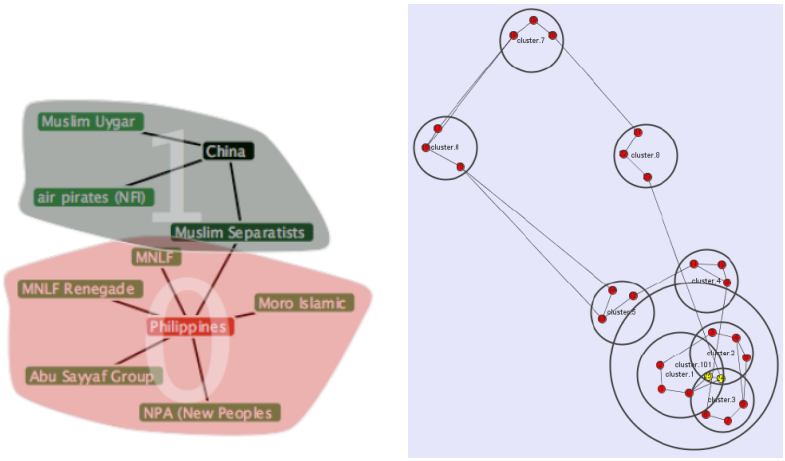
3 That is, in our context, 10 groups with at most 5 elements each one.

all the nodes) and then draws, in decreasing values of the z coordinate, additional graphs with lower levels of clustering, where nodes are clusters and edges join clusters that were together in the upper clustering.

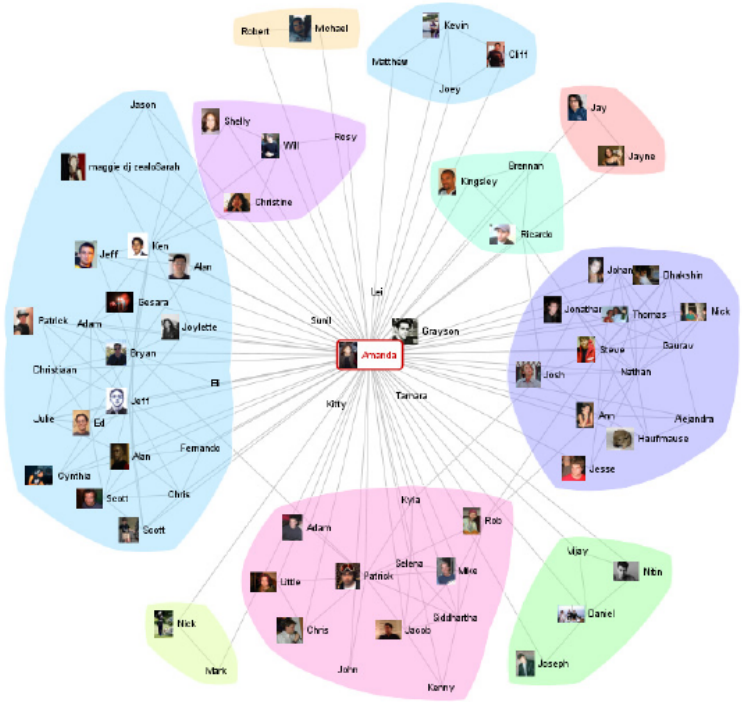
A *Compound Graph* [124] is a HCG in which the inclusion relationship serves to draw a hierarchical clustering within a single graph representation. The final visualization resembles a tree map [121].

Finally, the *Force-Directed Clustered Graph (FDCG)* is the most spread kind of clustered graph. A combination of repulsive and attractive forces for a single clustering are used in order to model inter-cluster and intra-cluster relationships. Sometimes, ancillary forces are also applied to simplify the structure or convey additional relationships, usually by means of dummy nodes. There are a number of social network tools that implement special FDCGs where the clustering is applied to an existing network, instead of defining the network (see fig. 37). For example, SocialAction [97] uses the Prefuse visualization kit [58] and central betweenness measures to determine and draw clusters. Vizster [57] is also based on Prefuse and group zones by clustering, allowing the user to define its granularity. It is usual in these implementations to display clusters from some level of a hierarchical clustering, allowing the change of this level (and therefore the clusters) at user's demand.

These tools are all examples of graph visualizations for groups, but none of them deal with overlapped groups. However, FDCG have recently been used to draw intersecting groups in a general approach [91]. This approach addresses exactly the problem discussed here, by means of the combination of a FDCG and several metrics that involve up to five additional dummy nodes per cluster. However, the shapes used to represent groups are very rigid (rectangles and circles) and it lacks of a strong visual design to enhance the understanding of group relationships and to deal with larger datasets (no more than 15 groups are represented, with low degrees of overlap).



(a) SocialAction (b) Omote and Sugiyama. [91]



(c) Vizster

Figure 37: Social network clustered graphs visualized by SocialAction (a) and Vizster (c). Omote and Sugiyama approach [91] to overlapping groups (b). Images from their corresponding papers.

9 VISUAL ANALYSIS ON BIOINFORMATICS

Visual analytics is a very recent branch of knowledge, originally focused on areas unrelated to bioinformatics. However, bioinformatics is plenty of information visualization approaches to several problems, and some of them actually point to visual analytics aspects. The following sections review some of the most relevant approaches towards a visual analytics application to bioinformatics problems.

9.1 COMPUTATIONAL INFORMATION DESIGN

Previously to the advent of visual analytics, the work of Ben Fry [49] already described a process schema that comprises most of the steps involved in the analytical process (see section 5.1).

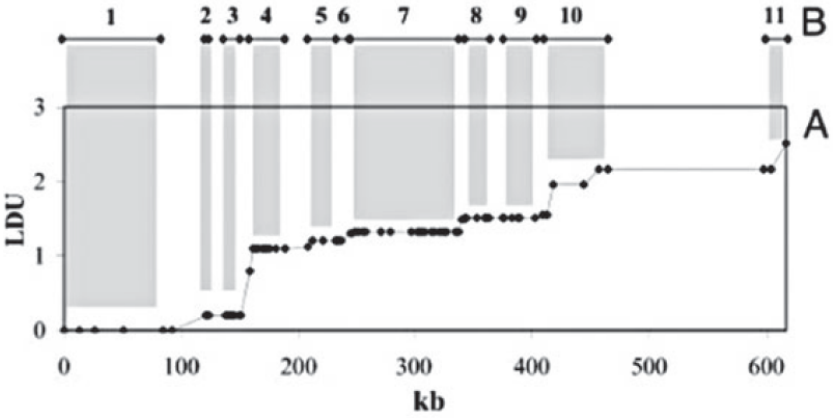
Moreover, Fry applied this model to several bioinformatics examples, specially on the field of genomics, such as the genome browsing or the SNP analysis[1]. In most of the discussed problems, Fry identified a lack of use of interaction and data mining in the visualizations. The improvements he proposed mainly are the use of data mining techniques previously to the representation of data and the implementation of interaction techniques in order to manipulate the representation. His approaches take into account that, eventually, the interaction with the tool can require to go back to previous steps such as filtering or mining.

As an example, fig. 38 illustrates the improvement of Ben Fry in the visualization of Linkage Disequilibrium (LD) units [147][2]. The static figure 38b only shows the improvements in the representation, mostly based on information visualization principles, such as the coding of the common/uncommon SNP sequences by color or the coding of percentages by rectangle heights and line widths. However, the implementation goes further: almost every visual entity can be interacted in order to allow the navigation through the LD units. In addition, some of these interactions imply a reprocessing of the information in order to filter some data or even to perform mining tasks, for example to recalculate the LD units by changing the parameter settings of the grouping algorithm that compute them. The visualization can also switch to other points of view, such as the quantitative traditional visualization of nucleotide bases and percentages, or the 3D visualization of the LD units. Fig. 38c shows the quantitative representation of LD units and the simple interface developed in order to change the point of view (top) and the thresholds for unit computation (bottom).

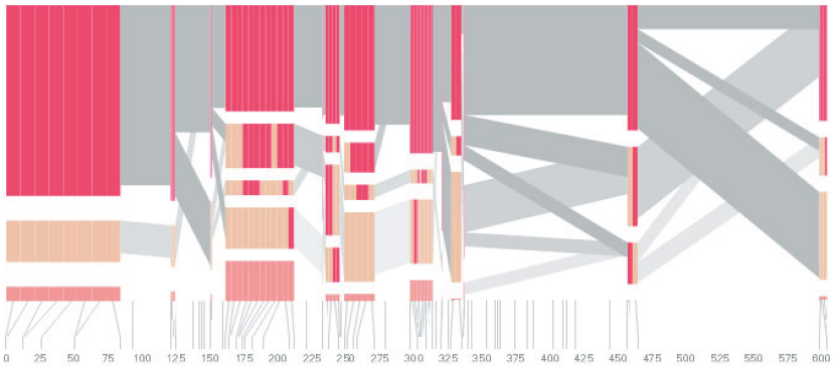
This visualization can be found at `benfry.com/isometricblocks`

1 A SNP is the variation of a single nucleotide when comparing two DNA sequences.

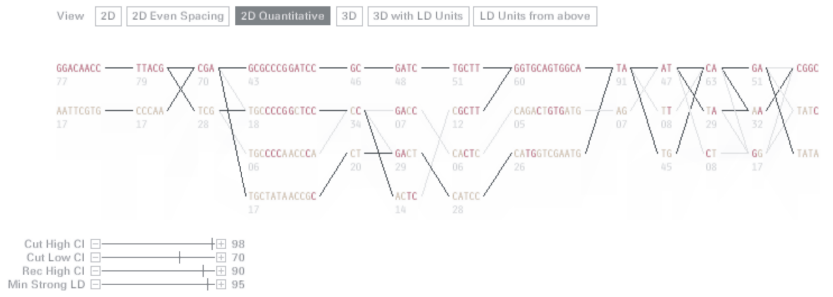
2 Between two DNA sequences of the same organism, there are typically several SNPs. Briefly, the LD describes the probability that a block of SNPs do not happen by chance. Therefore, a LD unit is a DNA sequence of SNPs that do not happen by chance.



(a) Zhang et al. LD map



(b) Fry LD map



(c) Fry LD quantitative map and interface

Figure 38: a) Zhang et al. representation visualizes LD units as rectangles and SNPs as points. b) Fry representation adds additional information in the form of color to represent the percentage of individuals with each SNP variation and lines that represent the linkage among an LD unit and the following one, and the percentage of individuals sharing them. c) Quantitative representation of b), and interaction options.

9.2 VISUAL ANALYSIS AND BIOINFORMATICS TOOLS

Apart for the original purposes of visual analytics (the analysis of security threats or disaster prevention), bioinformatics is probably one of the research areas with more tools oriented towards visual analysis.

Regarding gene expression analysis, Treeview [104] identifies visualizations with the main steps of the inspection process: overview, focus and details. Three main visualizations are displayed: the global dendrogram+heatmap, the zoomed heatmap and the gene annotations. The visualizations are linked so, for example, the selection of a branch of the dendrogram displays its zoomed portion of the heatmap and the annotations of its genes on the gene annotation visualization (see fig. 39). This reflects to some extent the gene expression analytical process:

1. Inspect the overview of the whole expression matrix
2. Classify the matrix into groups
3. Select a group, inspect the expression patterns
4. Review the available biological knowledge of the genes in the group.

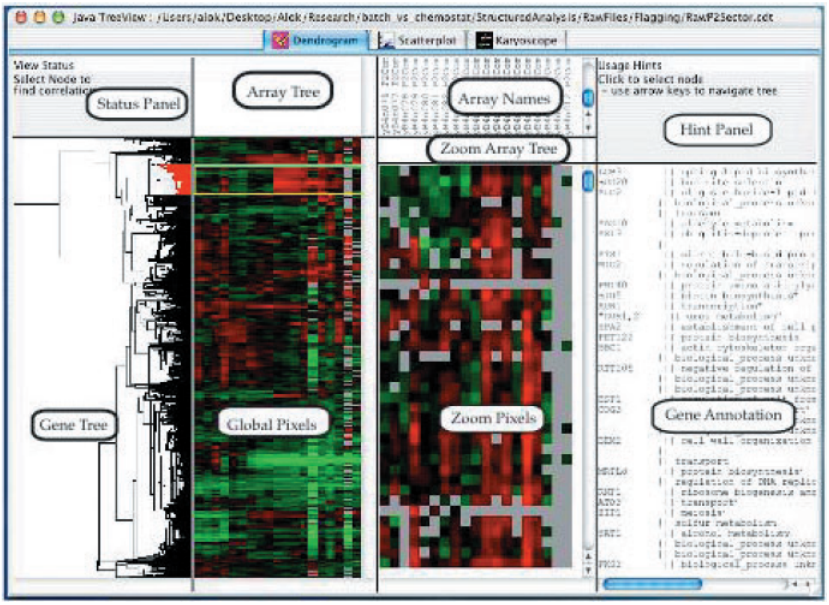


Figure 39: The original visualization of the heatmap is supported by additional visualizations for gene annotations and for the focused section of the microarray. Note that missing values are colored in grey. Reproduced from [104].

HCE [113, 114] is another example of the application of a multiple-linked views paradigm to the exploration of clustering results on gene expression data. It implements heatmaps, parallel coordinates, histograms and scatterplots in order to visualize expression data from different points of view (see fig. 40). Visual items (either representing gene or conditions) can be selected, modifying every visualization on real time. In addition, several clustering methods are available to apply to expression matrices. In the latest version, GO annotations can also be included in the analysis. This high degree of interactivity, visualization options and analysis techniques, within the single frame, make HCE be very close to the visual analytics on bioinformatics.

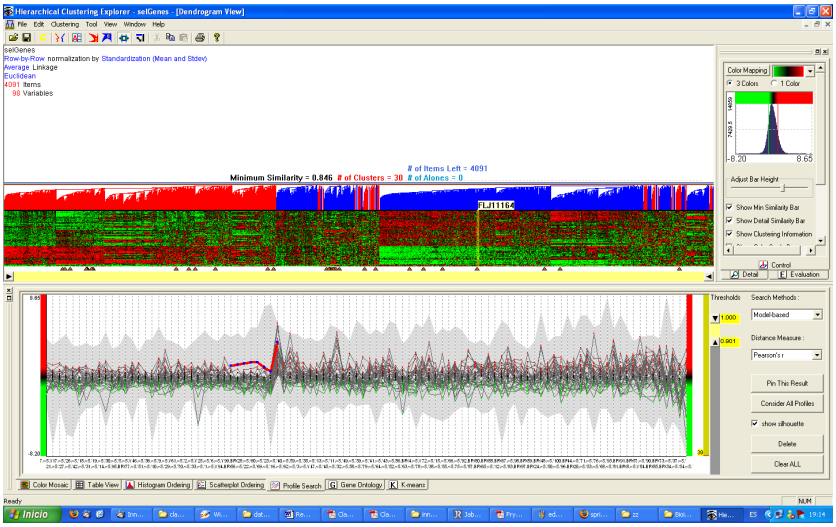


Figure 40: HCE visualization of a gene expression matrix and hierarchical clustering. At the top, a dendrogram+heatmap visualization with a threshold set to divide the matrix in 30 gene clusters. Below, a parallel coordinates visualization with a selection of gene profiles; the genes are also marked with triangles below the heatmap. At the right side, a histogram shows the expression level distribution.

On a different area, Hawkeye [110] is a more proper example of visual analytics applied to bioinformatics. On this approach, there is an intended analysis of the required steps for the inspection of genomes, specially the "finishing" step, which is the more time consuming and deals with the identification and correction of sequencing and assembly errors in a given genome[3]. The design of the interface is based on the information visualization mantra, allowing to go from the whole genome to interesting sections of the genome and to their details. The multiple-linked paradigm is also implemented by Hawkeye in order to visualize different statistics related to the genome (see fig. 41). However, Hawkeye goes beyond, implementing a proper visual analytics approach by integrating several analysis tasks (such as dynamic filtering and automated clustering) to focus attention and highlight anomalies in the genome. The authors are also aware of related available tools, such

3 Hawkeye also implements other secondary analysis tasks, such as the consensus validation of genes and the discovery of plasmids.

as the assembler tools; making their approach compatible with most of them. In order to fulfill the production and dissemination requirements of a visual analytics approach, the tool has a full user guide and open source code.

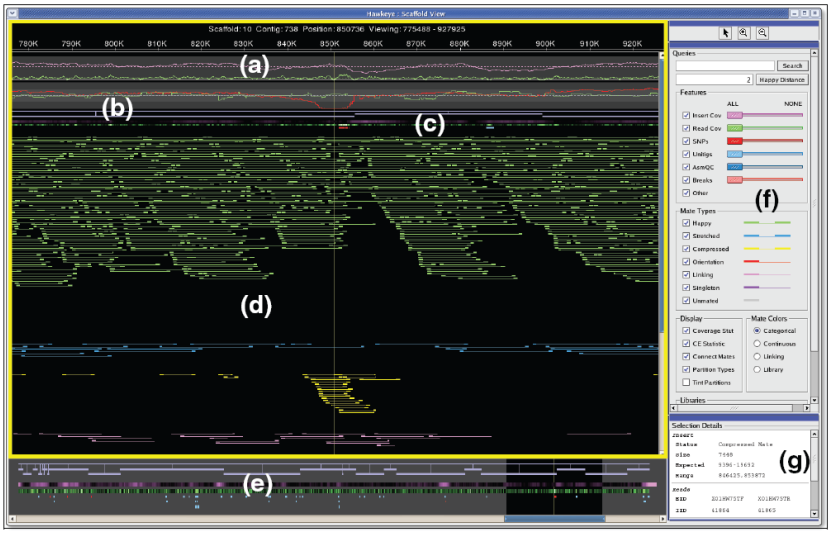


Figure 41: The scaffold view of Hawkeye provides a wide coverage of genome assembly. Up to seven visualizations are linked, five of them represent different quantities related to the genome sequence. Image found in [110].

Part IV

PROBLEM STATEMENT

10 PROBLEM STATEMENT

On this chapter, we identify the major problems involved with biclustering validation and visualization, discussing the limitations of the reviewed approaches. First, we enumerate the advantages and drawbacks of biclustering, concluding that its advantages are worth to try to solve its drawbacks. Second, we discuss the lack of use of internal indices and of the optimization of parameters for biclustering validation. Third, we discuss the limitations of bicluster visualizations when dealing with their special characteristics (grouping of genes and conditions, overlapping), focusing on the relevance of a proper representation of the overlappings. Finally, we discuss about the lack of visual analytics approaches for gene expression biclustering analyses and the advantages of designing it.

10.1 ADVANTAGES OF BICLUSTERING

The special characteristics of biclustering make it a better fitted technique for gene expression analysis than clustering or other similar non-supervised methods, because:

- A gene can be present in more than one biological function, and collaborate with different genes on each function. This is modeled by the ability to overlap biclusters.
- A similar behavior of two genes under a condition do not necessarily implies a similar behavior under other conditions. This is modeled by the two-dimensional (gene and conditions) grouping.

These advantages make biclustering outperform other techniques, specially clustering, when searching for functional groups of genes [84, 99]. In the future, the capability to generate experiments with more and more conditions, and the increase in the understanding of gene expression will make biclustering even more useful and necessary to interpret complex interactions among genes. This theoretical statement is confirmed for some biclustering algorithms on clustering/biclustering comparisons (see section 6.6).

10.2 DRAWBACKS OF BICLUSTERING

The strong points of biclustering are also its weak points to some extent. The flexibility of biclustering, specially the different definitions of what is a bicluster, makes it difficult to propose numerical benchmarks or somehow quantify goodness among biclustering methods.

On the other side, the fact that biclusters can overlap is difficult to covey on a visual representation. The visualization of overlapping groups is an open issue on information visualization, with techniques that successfully represent just a low number of groups (see chapter 8). The published visualizations of biclusters avoid this fact by replicating information (the BiVoc approach in section 7.2) or by oversimplifying

the problem (multidimensional scaling solutions such as the one of the gCluto tool in the same section).

The validation issues and the large amount of methods lead to a lack of standards for the use of biclustering (in opposition to, for example, hierarchical clustering for the traditional clustering). This makes the non-expert analyst to retreat from biclustering methods.

The visualization issues leads to the lack of representations that quickly give insight about the biclustering results. This makes the analysis of biclustering results uneasy, slow and too abstract compared to other methods, such as the dendrogram+heatmap approach for clustering.

In addition, small laboratories cannot perform large experiments with lots of experimental conditions because of its cost, so they usually deal with "slim" expression matrices with just a few columns (sometimes even just two). Therefore, there is little need of an analysis technique capable of grouping conditions. This circumstance will diminish due to reduction of microarray technology costs and the growing of public repositories that integrate several experiments (see section 2.2.5).

All these circumstances lead to a fact: a majority of the biclustering applications to gene expression analysis are done by their own authors. According to literature, there is little or no transition from biclustering design to biclustering application. Just an example: SAMBA [127] is a biclustering algorithm with over 200 citations[1], but most of them come from papers describing other biclustering algorithms or other non-practical papers such as surveys, tools, etc. This fact will probably change in the future with the formulation of more complex queries about gene relationships and the availability of larger gene expression matrices and easy-to-use, broadly validated biclustering algorithms and visualizations.

10.3 BICLUSTERING VALIDATION ISSUES

The widespread approach is to use non-biological external validation with synthetic data, and biological external validation with real data. Internal validation with real data is also very important, but it is frequently ignored (see section 6.5). Biological external validation metrics can complement but not substitute internal metrics in real data validation, because they have several flaws:

- They are biased to favor methods that find gene and/or condition relationships already known.
- Biological knowledge is not complete, so if the relationships inferred by a bicluster does not appear in our annotations, does it means that it is erroneous or it points to new, undiscovered information?
- Biological knowledge evolves quickly. For example, *Escherichia coli*'s TRN grew from 424 genes and 577 interactions in 2002 [118] to 1278 genes and 2724 interactions in 2004 [82]. Therefore, biological validation is unstable.
- Statistical significance tests, usually utilized in order to determine biological relevance, are controversial in statistical forums [7, 63].

1 following http://scholar.google.com

Besides, on comparisons such as the ones reviewed on section 6.6, the biclustering algorithms are usually tested with the initial parameter settings proposed by their authors. It is possible that this is not always the best setting, therefore undermining the performance of the algorithm. In clustering, relative indices are often used to find optimal initial parameters, but its use in biclustering literature is scarce, and only used to find stability when the algorithm has pseudo-random behavior [29], but not to find optimal initial parameters.

Also on these comparisons, gene-centered metrics should be avoided, at least for synthetic data, if no clustering algorithm is part of the comparison. If not, we are removing half of the biclustering grouping power (the grouping of conditions) from the validation. Furthermore, nowadays there is no special need to compare biclustering with clustering, because there are enough studies that confirm that a good biclustering algorithm outperforms clustering. For example, on the selected comparisons on section 6.6; Bimax, ISA or SAMBA outperformed clustering even without measuring the grouping of conditions, which favors clustering.

Finally, there are no compilations of biclustering algorithms, which could simplify the task of comparison and validation by third parties. The algorithms are usually available, but in different programming languages, sometimes without an open source.

The design of improved validation strategies is a vast task requiring of deep statistical knowledge and the testing of several use cases. Just as a start, on chapter 11, we propose the adaptation of a clustering internal index to biclustering and its use as relative index in order to determine the best parameter configuration. We also contributed to the development of a R package for biclustering algorithms.

10.4 BICLUSTERING VISUALIZATION ISSUES

Following the review in chapter 7, there are two widely accepted visualizations for single clusters: heatmaps and parallel coordinates. Both of them rely on the modification of representations of gene expression matrices, either by reordering the rows (heatmap) or by filtering them (parallel coordinates). The *visualization of single biclusters* can be successfully derived from single cluster visualization, just by applying another modification to the condition dimension of expression matrices. For example, regarding heatmap visualizations, BicAT reorders columns in the bicluster. Regarding parallel coordinates, some visualizations have been proposed for biclusters, but they can be improved. BicAT marks axes corresponding to conditions in the bicluster, but it is hard to visualize the bicluster as a whole (see fig. 29a). The visualization of BiVisu filters the axes that represent conditions out of the bicluster, which facilitates the visualization of the bicluster as a whole but misses the context of the rest of conditions (see fig. 29b).

The *visualization of multiple biclusters* is harder. Because of the property of overlapping, just to reorder the rows and columns of the heatmap representation is not enough if the degree of overlap is moderately high (see section 7.2). We reviewed an approach [54] that replicates the rows and columns that are in several biclusters but, although the algorithm minimizes the number of repeated rows and columns, this replication can lead to ambiguities and misinterpretations. For example, the tall bicluster at the top of the left matrix of fig. 32, actually overlaps in a condition and several genes with the two biclusters at its right, but

to perceive it you must inspect the repeated condition names/profiles and infer the relationship. We face similar problems when we try to represent several biclusters with parallel coordinates. In this case, the geometrical limitations are higher, and unless the biclusters (or even the clusters) are very separated, the representation becomes rapidly cluttered. Another option is to forfeit the exact representation of expression levels and deal with biclusters as entities on their own. gCluto [101] applies this by means of multidimensional projections of clusters. However, the application of this technique to biclusters do not satisfactorily represents overlap: the multidimensional scaling oversimplifies the problem. In addition, there is no linking to the elements inside the biclusters which impedes a detailed inspection (see fig. 33).

Therefore, the visualization of biclusters is still open. Following the reviewed options, the solution may be in the representation of biclusters as independent visual entities, but conveying overlap more precisely. The visualization of set diagrams (section 8.1) is a good option to represent overlapping groups, but the geometrical limitations are too strict to allow the representation of several biclusters (for example, more than 10). Clustered graphs (section 8.2), specially FDCG have been successfully applied to non overlapping groups.

On chapter 12 we present a novel visualization technique for the representation of biclusters based on a FDCG-like model. This technique properly conveys overlap without replicating or oversimplifying the information. However it will forfeit expression levels in order to achieve it, so it will require of additional linked gene expression visualizations.

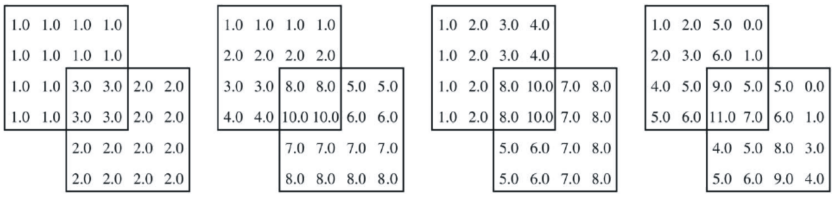
10.4.1 *The Relevance of Overlap*

Overlap is an intrinsic characteristic of biclusters but, is it relevant for the interpretation of biclustering results, is it worth of designing a visualization technique that properly conveys it? The answer is yes, at least in the two mayor ways we describe in this section.

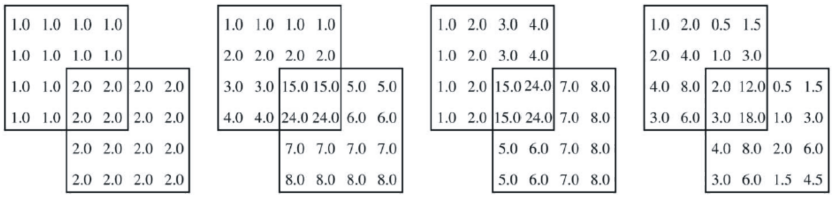
Structural Interpretations of Overlap

Biclusters describe how data are structured. One of the main assumptions of biclustering is that the structure can consist of a superimposition of local structures. Therefore, two local structures (for example, two groups of objects representing variables with high values for different conditions) can intersect one another, either on objects, conditions or both; combining their structures at the intersection. This point of view is difficult to model with traditional clustering, which was not designed to search for local structures (regarding to conditions) or to deal with intersections.

Madeira and Oliveira [84] identify two main ways of overlap, defined by the general additive and multiplicative models (see fig. 42). More complex overlaps can be defined (for example, the average) but usually the additive and multiplicative models are applied (see section 6.2). A small example, conveying the incomes of some companies due to patent and product sales, is illustrated in fig. 43. Traditional clustering (in blue) should find three clusters $(c1, c2), (c3, c4), (c5, c6)$, which identify the companies with the same profiles through all conditions, but there is no inference of other relationships such as the fact that $c1, c2, c3$ and $c4$ have the same product incomes. An additive biclustering method



(a) General additive model



(b) General multiplicative model

Figure 42: Intersection models for (from left to right) constant, constant by rows, constant by columns and coherent evolution biclusters. Note that the intersections are of the same type than the overlapped biclusters (reproduced and corrected from [84]).

could find the biclusters colored in red, (c1, c2, c3, c4, product, total) and (c3, c4, c5, c6, total, patent). The ability of biclustering to overlap biclusters made it possible to classify the same companies in more than one group, thus properly discovering the groups of companies profitable on products and profitable on patents. Moreover, the overlapping area by itself identifies the group of companies that, regarding the total incomes, are the most profitable.

company	product incomes	total incomes	patent incomes
c1	100	100	0
c2	100	100	0
c3	100	200	100
c4	100	200	100
c5	0	100	100
c6	0	100	100

Figure 43: Example of clustering and biclustering.

The meaning of an overlapping region varies depending on the application field, but it often refers to consensus, combination of characteristics, collaboration about functional processes, etc. It is a good analytical process to start with the highly overlapped zones and continue with more separates zones, so we can identify during the process the main functionalities and how they relate. Sometimes these overlapping structures have relevant information, and sometimes they only convey the structure that is obvious. Regardless of the case, it is important to detect them in order to focus on or discard them.

Considering the objects and conditions grouped by biclustering, the overlapping areas give rise to what we called *superbiclusters*. Follow-

ing the notation in section 6.1, a superbicluster of order k (S_k) is the overlapping matrix of k biclusters:

$$S_k = O(B_1, \ldots, B_k) = A(G_1 \cap \ldots \cap G_k, C_1 \cap \ldots \cap C_n) \tag{10.1}$$

For example, the intersection among the two biclusters of fig. 43 is a S_2 superbicluster containing $(c3, c4, \texttt{total})$. S_k^e is an *exact* superbicluster if the genes and conditions in S_k^e are not included in any bicluster different from $B_1, \ldots, B_k$.

Biological Interpretations of Overlap

Talking about gene expression, an intersection of two biclusters can refer, for example, to a group of genes that under certain conditions are regulated by two transcription factors, or involved in two biological processes. In these cases, it seems appropriate to use an additive model, because the transcript abundance of both processes is expected to be aggregated. Superbiclusters of very high order may refer to transcription factors that are present under different conditions or biological functions, so probably they are not as much interesting as other structures, such as superbiclusters of lower order or exact superbiclusters of order one[2]. Other overlapping structures can give way to different interpretations, for example superbiclusters of order 2, even if they contain only one gene, can convey bridges among two biological processes. section 15 shows some of these structures on real cases.

We have studied GO enrichment[3] of superbiclusters and exact superbiclusters and they do not improve the enrichment of normal biclusters (see table 7). However, specially in biclustering algorithms that implement exhaustive searches, overlap can be considered as a measure of the *effect size*. Adapted to biclusters, it means that the more times two elements are grouped together in a bicluster, the more tight is their relationship. Therefore, if several biclusters back up a group of genes with GO enrichment, the resulting superbicluster is not only statistically significant but strong in effect size.

The effect size is a measure of the strength of the relationship between two variables, by counting the number of evidences that support this relationship.

RESULT SET	NUMBER OF GROUPS	% of groups with at least one GO term under p-value			
		0.1	0.01	0.001	0.0001
all biclusters	363	100.0	90.9	29.2	7.7
all S_k, $k \geqslant 5$	132	99.2	80.3	15.9	3.8
all S_k, $k \geqslant 10$	83	98.8	77.1	9.6	2.4
all S_k, $k \geqslant 50$	2	100.0	50.0	0.0	0.0
all S_k^e	170	98.8	81.2	19.4	3.5

Table 7: GO enrichment on normal biclusters, superbiclusters and exact superbiclusters after Bimax biclustering of Eisen et al. microarray experiment [44].

2 These biclusters group genes or conditions that are grouped just by one bicluster, so they are clearly separated in the expression data matrix

3 See section 6.5

10.5 VISUAL ANALYSIS OF GENE EXPRESSION DATA

The number of pure visual analytics approaches on bioinformatics is still scarce. This is mainly because it is a very novel research field, but it is included within the top ten challenge areas of visual analytics [76]. However, as reviewed on section 9.2, there are examples of approaches that implement several visual analytics principles, specially the identification of analysis phases, the iteration among these phases and the high interactivity. In addition, the use of multiple-linked views proved useful in order to visualize several approaches to the same data. On the other side, there are some drawbacks in these and other tools for analysis and visualization on bioinformatics that must be taken into account for developing a visual analytics approach to the problem:

- *Data heterogeneity*: due to the variety and complexity of biological studies, the data sources are also diverse. For example, a gene can be seen or represented as a sequence, as a protein producer, as part of a biological pathway, as a set of probes in a microarray, etc. Each of these scopes relates to a different research area involving different laboratories, experiments, etc. However, all of them are useful in gene expression analysis, but even with the best of the efforts, to integrate all these data is complicated.

- *Speed of change*: biological data are not only heterogeneous but also they change quickly: gene sequences are corrected, new gene functions are discovered, etc. Therefore, it is important to request for renewed input periodically, or to connect to data sources in real time[4].

- *Monolithic visualizations*: biologists and bioinformaticians are familiar with some visualizations, such as heatmaps, dendrograms, phylogenetic trees or genome browsers. Furthermore, sometimes they are used to particular configurations of these visualizations, such as green-black-red scales for microarray heatmaps. These visualizations are not always the best fitted, and usually additional, novel visualizations help in the visual analysis process but, on the other side, too many innovations could lead to an information overload for the analyst. Novel techniques must be easy to use and attractive, and traditional visualizations should be available so the user doesn't feel lost.

- *Time performance*: visual analysis tools require a real time interaction. However, the dimension of data and the complexity of analysis techniques usually undermine time performance, also downgrading interaction. Performance bottlenecks should be identified and optimized, or otherwise treated (progression bars, warning messages, etc.) in order to keep the user in control of the tool.

- *Formats*: several bioinformatics tools require complex formats. It is optimal to use standard formats or, if not possible, simple formats that can be easily transformed to other formats. The amount of time spent switching formats is not trivial, and any savings are important.

4 For example, the original probe-to-gene mappings of Affymetrix chips should be revised if the sequences related to genes change. If not, the transcription of incorrect probes are being used to calculate the transcription of the gene

We have kept this potential pitfalls in mind when designing our solution. In addition, because of the novelty of visual analytics, the examples discussed in section 9.2 are usually not completely based on a proper visual analytics design. Our approach takes profit of the advantages of the reviewed examples and the now available visual analytics paradigms (see section 5.1) to implement an approach to gene expression visual analysis by means of biclustering.

Part V

PROPOSED SOLUTION: DESIGN

11

EXTERNAL AND RELATIVE INDICES FOR BICLUSTERING VALIDATION

In this chapter we present a proposal to apply relative and internal validations of biclustering results on gene expression. Usually, external indices are used to measure the goodness of biclusters, either by means of synthetic biclusters embedded in test matrices or by means of available biological knowledge. Unfortunately, external indices present a number of drawbacks, as previously discussed (see sections 6.5, 10.2). The use of internal indices is desirable to solve some of these drawbacks, but also they are useful on their own. The first section defines the adaptation of a well-known cluster validation index for bicluster validation, the Hubert statistic. The second section describes an experiment to test the use of this adapted statistic as a relative index in order to improve the parametrization of biclustering algorithms, discussing the results of its application to a simple example.

11.1 ADAPTATION OF THE HUBERT STATISTIC TO BICLUSTERING

The Hubert statistic (Γ) is a measure used for statistical cluster validity ([70], page 148). It measures the correlation among the entries of two $n \times n$ matrices, X and Y, on the same n objects. x_{ij} denotes the observed proximity between objects i and j, given a distance measure. In clustering, y_{ij} is often defined as zero if objects i and j are in the same category or cluster and one otherwise. The normalized Γ statistic ($\bar{\Gamma}$) is:

In order to use these matrices for internal validation, they must contain data without built-in or known relationships.

$$\bar{\Gamma}(X,Y) = \frac{\frac{1}{k}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(x_{ij}-\mu_X)(y_{ij}-\mu_Y)}{\sigma_X\sigma_Y} \tag{11.1}$$

where $k = n(n-1)/2$, μ_X and μ_Y are the means of the matrices and σ_X and σ_Y are their variances.

The adaptation of the Γ statistic to biclustering needs to address its two main characteristics: bi-dimensionality and overlap.

About bi-dimensionality, in order to capture correlation among genes *and* among conditions, two pair of matrices are defined, (X^g, Y^g) for genes (rows) and (X^c, Y^c) for conditions (columns), thus computing two indices, $(\bar{\Gamma}^g, \bar{\Gamma}^c)$. In order to obtain the adapted Hubert statistic ($\bar{\Gamma}'$), both indices are then combined by means of a weighted mean:

$$\bar{\Gamma}' = \frac{n\bar{\Gamma}^g + m\bar{\Gamma}^c}{n+m} \tag{11.2}$$

where n is the number of genes and m the number of conditions. Proximity matrices X^g, X^c are computed with the Euclidean distance from the input expression matrix A, just as in the case of clustering:

$$x^g_{ij} = \frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sqrt{\sum_{k=1}^{m}(a_{ik}-a_{jk})^2} \tag{11.3}$$

$$x^c_{ij} = \frac{1}{m} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \sqrt{\sum_{k=1}^{m} (a_{ki} - a_{kj})^2} \tag{11.4}$$

To address overlap, the clustering matrix Y is substituted for the biclustering matrices Y^g, Y^c, with

$$y_{ij} = 1/(1 + k_{ij}) \tag{11.5}$$

where k_{ij} is the number of biclusters in which the object i (a gene in Y^g, a condition in Y^c) is grouped together with the object j. y_{ij} is in $(0, 1]$, being one if they are never grouped together and going down to zero if they are grouped together in several biclusters.

The $\bar{\Gamma}'$ statistic measures the degree of correspondence between the entries of X and Y, therefore measuring to some extent the fitness of the biclusters to the structure of the data.

We can introduce variations in the statistic by changing the way of building X and Y matrices. X matrices can be computed by means of any other distance measure, while Y matrices can use any other formula to measure the degree of relationship among features.

See section 6.5 for an enumeration of validation indices

It must be noted that the $\bar{\Gamma}$ index and similar indices, such as PCCC, are less precise than external indices. For example, [70] lists different drawbacks of the PCCC, estimating that even a value of 0.9 (PCCC values are in $[0, 1]$) would not be enough to assert that there is a good correlation between X and Y. The differences between external and internal indices are similar to the differences between supervised and non-supervised classifications. The former is more accurate but requires additional knowledge; the latter is less precise but does not require *a priori* information.

11.2 APPLICATION OF RELATIVE INDICES TO BICLUSTERING

Relative indices are typically used to determine the best choice of parameters of an algorithm for a particular data set. Authors usually propose a "best parametrical" setting for their algorithms, and it is usually applied in third party comparisons [99, 90] and by most of the users. However, this can undermine the performance of the biclustering methods that are very sensitive to the nature of data and, at any rate, it should be interesting to compare algorithms with the best parameter configuration for each data set.

Typical parameters for biclustering are the maximum number of biclusters, the minimum number of genes or conditions to include, the minimum similarity among profiles, etc.

Relative indices can be made of external or internal indices. External indices are good to compute the best parameters of biclustering algorithms for a synthetic dataset. However, internal indices must be used to search for the best parametrization on unknown data.

Independently of the kind of index, the procedure is to run the biclustering algorithm with different parameter settings, and then to compute the index for each one. The parameter setting with the best index is selected as optimal for the data set. The selection of the ranges of parameters to include in the procedure should broadly represent different combinations of their values.

Our idea is to apply the proposed $\bar{\Gamma}'$ statistic and the F1 measure [134] to the identification of the best parameter configuration of a given biclustering algorithm.

The F1 measure defines the matching between two biclusters. Suppose that we want to compare a bicluster A, known to be in the matrix, and a bicluster B, found by a given algorithm. Let g_X be the number of genes in X, c_X the number of conditions in X and $n_X = g_X s_X$ the number of expression levels in X. Sensitivity, specificity and F_1 are defined as follows:

$$sensitivity(A, B) = \frac{g_{A\cap B}}{g_B} \times \frac{c_{A\cap B}}{c_B} \tag{11.6}$$

$$specificity(A, B) = \frac{g_{A\cap B}}{g_A} \times \frac{c_{A\cap B}}{c_A} \tag{11.7}$$

$$F_1(A, B) = \frac{2(g_{A\cap B})(c_{A\cap B})}{n_A + n_B} \tag{11.8}$$

Sensitivity measures the proportion of genes and conditions in B that are also in the embedded bicluster A. Inversely, specificity measures the proportion of A that is also in B. The F_1 measure is the harmonic mean of the sensitivity and specificity.

Following the process defined by [99], we can use the F1 measure to compare the matching between two biclustering result sets (not only two single biclusters). Let E be a set of e biclusters embedded in the synthetic matrix, and R a set of r biclusters obtained by a given biclustering algorithm. We can calculate the averages of the maximum F_1 match scores:

$$S_F(E, R) = \frac{1}{e} \sum_{E_i \in E} max_{(R_j \in R)} F_1(E_i, R_j) \tag{11.9}$$

$$S_F(R, E) = \frac{1}{r} \sum_{R_j \in R} max_{(E_i \in E)} F_1(R_j, E_i) \tag{11.10}$$

A: the gene expression matrix
E: the known bicluster set
P: set of parameter settings for the biclustering algorithm
Calculate X^g, X^c, distance matrices for genes and conditions in A
for each parameter setting p_i in P **do**
  Run the biclustering algorithm with p_i parameters, obtaining the result set R_i
  Calculate $S_F(E, R_i)$ and $S_F(R_i, E)$ following eqs. 11.9, 11.10
  Calculate $SS_i = mean(S_F(E, R_i), S_F(R_i, E))$
  Calculate matrices Y^c, Y^g from R_i following eq. 11.5
  Calculate $\bar{\Gamma}'(X^c, Y^c)$ and $\bar{\Gamma}'(X^g, Y^g)$
  Calculate $\bar{\Gamma}'_i$ following eq. 11.2
end for
Select the p_i corresponding to the highest SS_i as the optimal parameter setting. Mark its index as i1
Select the p_i corresponding to the highest $\bar{\Gamma}'_i$ as the optimal parameter setting according to the adapted Hubert Statistic. Mark its index as i2
Calculate $\bar{SS}$, the average mean of sensitivity and specificity for all the parameter settings.
Store p_{i1}, SS_{i1}, p_{i2}, SS_{i2} and $\bar{SS}$

Figure 44: Algorithm to find optimal biclustering parameters

$S_F(E, R)$ is the *average module recovery* of the biclustering algorithms for the embedded biclusters, that is, the capacity to retrieve known structures from the expression matrix. This is similar to the specificity concept. $S_F(R, E)$ is the *average bicluster relevance* of the biclustering algorithm, that is, the sensitivity or capacity to retrieve just the known structures and not additional spurious biclusters (false positives). The average of these two values (SS) represents the overall specificity and sensitivity of the biclustering algorithm for a given parameter setting.

The best parameter setting is the one with the bicluster set that got the highest SS, which is the best solution we can get with the biclustering algorithm. The parameter setting corresponding to the results with the highest $\bar{\Gamma}'$ should give a suboptimal solution providing we do not use a priori information. Then, we check the SS of this suboptimal solution, comparing them with the embedded biclusters. Finally, we compute and store the average SS of all tested parameter settings. The SS values with F_1 will be the best ones, but the SS values with $\bar{\Gamma}'$ should be better than the average SS. The whole process is summarized by the algorithm in fig. 44.

12 VISUALIZATION OF BICLUSTERS

In chapter 6 we described the special group characteristics of biclusters: bi-dimensionality and overlap. The visualization of these characteristics is a difficult task just using techniques designed for clustering. Either they cannot be adapted because of geometrical limitations, or the adaptation does not properly convey overlap (see section 10.4). In this section we discuss an approach inspired by Euler diagrams and clustered graphs to visualize several biclusters that accurately conveys overlapping and enhances the analysis of biclusters.

12.1 OVERLAPPER

Overlapper is the name for the proposed visualization technique for the representation of overlapped biclusters. The main objectives of Overlapper are:

- Display more than ten biclusters, with arbitrary number of elements and overlapping degree; within the frame of subset standard.
- Keep both levels of information (elements and biclusters) available to be visualized in the same display, to avoid losing context.
- Do not simplify or duplicate information. Both approaches could give clearer visualizations, but at the cost of losing information or adding ambiguities.
- Boost the identification of subgroups of elements found together in several biclusters (superbiclusters).
- Allow the representation of bicluster results from different sources in order to visually compare them.
- Provide ways of interaction to enable different points of view and to facilitate the exploratory analysis.

In order to achieve these objectives we chose a representation based on a FDCG. The proposed technique is valid for any kind of overlapping groups, although the main purpose of overlapper is to represent biclusters on genes and conditions[1].

1 We discuss some of these additional applications in section 15.4

12.2 GRAPH MODEL

Because spatial position is one of the most relevant characteristics for perception, the choice of a graph model to locate nodes in the display is a key factor[2]. It is our goal to have elements pertaining to exactly the same biclusters together, elements coincident in some biclusters relatively close and elements in completely different biclusters separated.

Force directed graph models have been successfully used by several authors for networks of medium size. It allows control over the placement of nodes by the definition of edge connections and edge lengths, and the setting of force strengths. As a drawback, these graph visualizations usually present edge cluttering when the number of nodes and edges grow. We reduce this cluttering by hiding edges and drawing hulls instead to represent biclusters (see fig. 45).

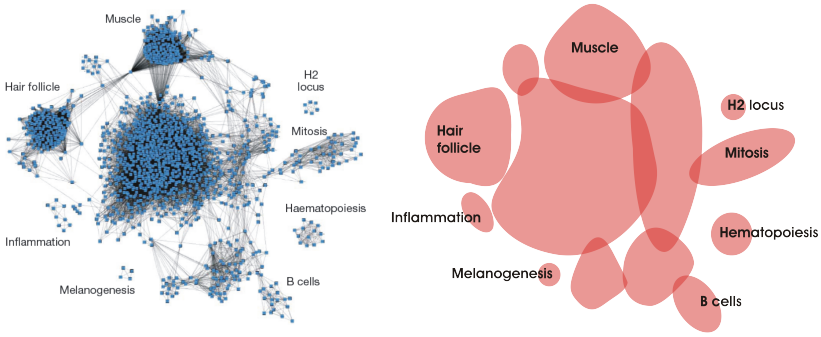


Figure 45: Simplification of a mouse skin tumor network (left) by means of hulls (right). Left figure found in [100].

Following the definitions of section 6.1, let $B = \{B_1, B_2, ..., B_n\}$ be a set of biclusters, where bicluster B_k contains genes G_k and conditions C_k. We refer to genes and conditions indistinctly as elements, so B_k contains elements $U_k = G_k \cup C_k$. Let $U^T = G^T \cup C^T$ be the set of all the elements. Let $U' \subseteq U^T$, and let $f(U')$ give the set of biclusters $B' \subseteq B$ that contain *all* the elements in U'.

To represent the biclusters in B, we define a graph as a pair of sets (E, V), being E the set of edges and V the set of vertices of the graph. We chose two different methods to build the graph:

- *Complete subgraphs:* For each element $u_i \in U^T$ that is at least in one bicluster, add a vertex v_i to V. For each bicluster B_k of n_k elements, with corresponding vertices V_k, add the subset of edges $E_k = \{e(v_1, v_2) : v_1, v_2 \in V_k\}$ to E. This is equivalent to add to the graph the corresponding K_{n_k} complete graph[3] for each B_k (fig. 46b).

- *Complete dual graph:* Let $Z = \{Z_1, ..., Z_p\}$ be the set of zones in B, so the elements in Z_k are *exactly* in the same biclusters[4], that is,

2 Proximity is also a powerful feature in order to detect groups

3 A complete graph of n nodes (K_n) is a graph where every node is connected to every other node in the graph

4 Note that zones correspond to the geometrical definition of exact superbiclusters

$f(u_1) = f(u_2) = f(Z_k)$. For each zone Z_k add a vertex z_k to V. For each pair of zones (Z_i, Z_j) containing elements that share one or more biclusters ($f(Z_i) \cap f(Z_j) \neq \emptyset$), add an edge $e(z_i, z_j)$ (fig. 46d).

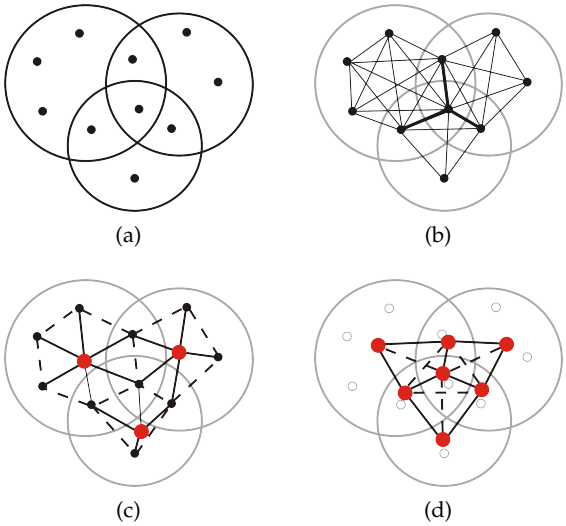


Figure 46: a) Abstract groups and elements. b) Complete subgraphs with contours and vertices corresponding to the abstract diagram in a. c) Radial model for the same example (wheel graph with dashed lines). Red dots represent dummy nodes added for each group d) Dual graph (complete dual graph with dashed lines). Red dots represent dummy nodes added for each zone (also called dual nodes).

Note that a complete dual graph has the same vertices than a dual graph, but it builds complete subgraphs for the nodes involved in each group. The reason to build the edges in such a way is to reinforce the group structure. Because of it, other building methods have been discarded, specially tree and radial methods [17]. Although these methods reduce the required number of edges, it is done at the cost of losing group cohesion. For example, the radial method adds dummy nodes for each bicluster, and then connects every node in the bicluster to the corresponding dummy nodes, but keep them unconnected with each other (see fig. 46c), permitting a separation of a double edge length for them or, on the contrary, to be very close, forming narrow and elongated shapes when forces are applied. This can be solved by adding peripheral edges (dashed lines), in a 'wheel-like' model, but this will favor some nodes to be closer than others in the same group, and it will double the number of edges.

The complete dual graph (from here on, we will refer to it just as dual graph) is a simplification of the complete subgraph model, which improves computation-time performance and reduces edge cluttering, at the cost of relaxing the structure. We identified four major factors that determine the computational complexity of the graph, that will also affect to its visual complexity:

- Number of elements ($|U|$)
- Number of biclusters ($|B|$). Usually $|B| \ll |U|$.
- Average number of elements in a bicluster ($\overline{|B|}$) and in a zone ($\overline{|Z|}$).
- Number of zones ($|Z|$, $|Z| \geqslant |B|$). Usually $|Z| \ll |U|$. It depends on the average overlapping degree among biclusters: the higher the overlapping, the larger the number of zones.

A complete subgraphs model has $|U|$ nodes and up to $|B|\overline{|B|}(\overline{|B|}-1)/2$ edges (a bit less if we do not repeat shared edges in the intersections, represented as bold lines in fig. 46b). A dual graph presents only $|Z|$ nodes and up to $|Z|\overline{|Z|}(\overline{|Z|}-1)/2$ edges. Except in the cases where the overlapping degree of groups is very high, $|Z| \ll |B|$, and therefore the dual graph is much simpler. For a force directed layout, the complexity is of $O(n^3)$, being n the number of nodes in the graph [59]. Therefore, regarding algorithmic complexity, the dual graph is the best option. Regarding edge cluttering, again the dual graph is simpler, being edge crossing extremely high for a complete graph, even for simple cases (see fig. 46b). However, because of the additional edges, the complete subgraphs model has a more robust structure that in some cases conveys better group relationships. Both methods were implemented, allowing through interaction to switch between them.

After building the graph, a force directed layout is applied, as described in [48]. This method will separate unconnected nodes by means of an expansion force and will keep connected nodes close by means of spring forces. Expansion forces are applied among every node, while spring forces are applied only among connected nodes. The only parameter that varies depending on the graph model is the stiffness of spring forces, which is weighted by a factor. In the case of the complete graph, edges shared by n groups have a weight factor of n (in fig. 46b, bold edges will have a factor of 2, the rest of 1). In the case of the dual graph, an edge between two dual nodes has a weight proportional to the number of groups shared by them.

12.3 VISUAL ENCODING

The discussed graph model is a simple yet powerful way to visualize groups and elements in the groups, granted by the Gestalt law of proximity. In order to improve it, we added several visual encodings for biclusters and elements.

Regarding biclusters, they are represented in the graph by complete subgraphs, but their edges are not drawn unless requested by user, because although connectedness is a powerful grouping principle, edge cluttering easily occurs with large graphs [59]. We substituted them for contours wrapping all nodes within each bicluster, drawn as simple closed curves (*hulls*). We selected this kind of curves because they have minimal perimeter, to reduce contour cluttering and maximize continuity seeking. With this representation we make profit of the perceptual grouping factors of closure and continuity[5] to represent biclusters. To draw each hull, the position of the outermost nodes of each bicluster are used anchors for a closed spline curve. The area enclosed by each contour is filled with a transparent color, with the same hue for all contours to avoid color cluttering. The use of transparent colors make the intersecting areas solider, facilitating the detection of overlaps (see fig. 47).

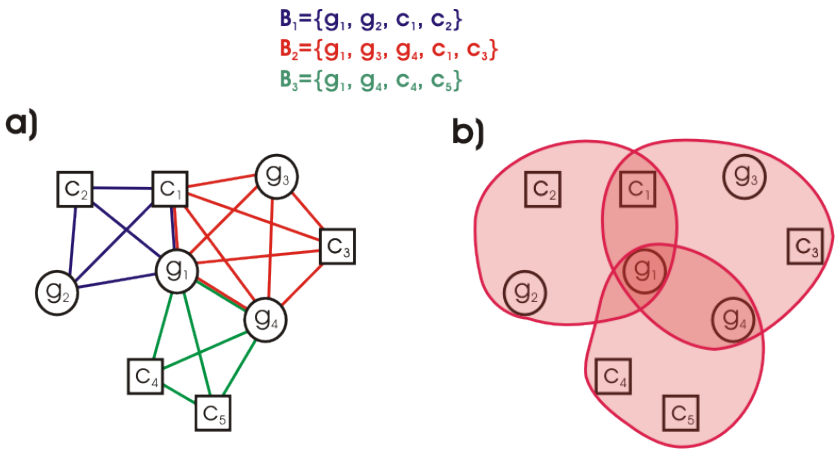


Figure 47: Overlapper graph structure.

Regarding the grouped genes and conditions, they are visualized as simple unfilled circles and squares, respectively, as in fig. 47. The nodes in a zone can be merged into a single dual node, with an area proportional to the number of elements in the zone (see fig. 49f). Piecharts superimposed to these simple shapes represent the number of biclusters the node belongs to (the pie is divided into as many sectors as biclusters). This way, it is easy to quantify the number of biclusters the node belongs to, at least up to 6-8 groups (for a simple case, see fig. 49b). The different shape of piecharts helps to identify and visually group intersections and superbiclusters, enhancing the Gestalt law of proximity with the law of similarity.

Although the graph model and the visual encoding conveys satisfactorily elements, biclusters and their relationships, our approach it is not

5 See section 4.2.4

exempt of drawbacks. Like any visualization of groups with contours, it is very difficult if not impossible to visualize a large number of groups without relaxing restrictions (see section 8.1). These relaxations can give way to empty zones and elements surrounded by groups they are not in (see fig. 48).

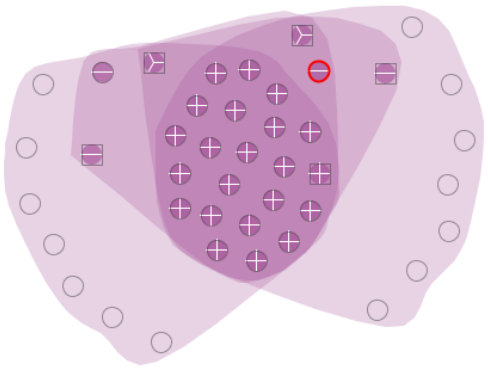
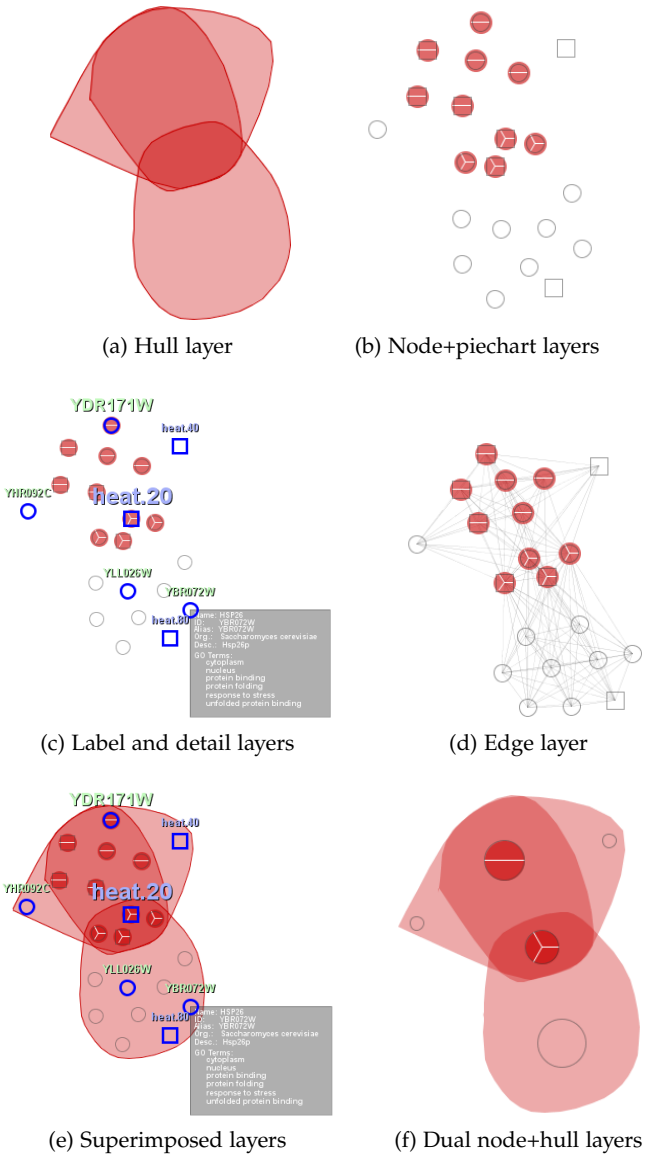


Figure 48: The node highlighted in red is misplaced. It is in an area surrounded by three biclusters but actually it is only in two biclusters. In this case, its piechart helps to detect the misplacement: the node should be on the area at its right, with the other two-sector piechart.

In order to minimize these undesirable effects, an option is to define heuristic layout metrics such as the ones described in [47, 91]. For example, a metric can calculate the number of nodes that fall on the area of a bicluster they are not in, and move them out. More specific metrics can be defined following particular relationships among groups, maybe adding additional edges to the graph model or modifying the edge forces on special cases. On our experiments with metrics, generical metrics usually distort too much the graph representation, add large computational loads and give way to unaesthetic visualizations. Specific metrics have better computational and visual performance, but they tend to be highly data-dependent and may lead to contradictions if used exhaustively.

In line with the visual analytics approach, we have chosen another option: to rely on interaction and the superimposition of layers to sort out any possible ambiguity. The visual encodings are distributed in layers that can be superimposed without occlusion. The layers are (see fig. 49):

- *Node layer*: nodes are drawn as simple, transparent shapes.
- *Piechart layer*: nodes are drawn as transparent piecharts.
- *Hull layer*: biclusters are drawn as transparent areas with solid contours wrapping grouped nodes.
- *Label layer*: names of nodes and biclusters.
- *Detail layer*: textual information of nodes, if available (for example, in the case of genes: definition, organism, GO terms, etc.).
- *Edge layer*: the underlying edge structure is drawn.



(a) Hull layer (b) Node+piechart layers

(c) Label and detail layers (d) Edge layer

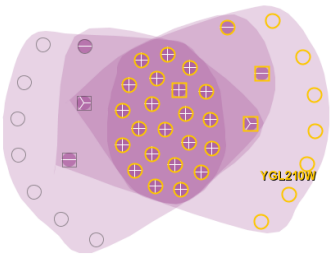
(e) Superimposed layers (f) Dual node+hull layers

Figure 49: Different visualizations of Overlapper for three biclusters.

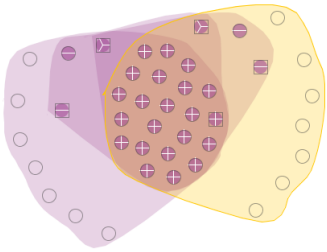
The main layers (node, piechart and hull) make use of transparency, so they can be superimposed in any order. The superimposed layers must have different perceptual characteristics in order to avoid confusion among layers [141]. In our design, node/piechart and hull layers are clearly different because of the dimension of the areas they represent (the Gestalt law of relative size). It is not very important if the node and piechart layers are identified as the same layer because both refer to the same element. Problems could arise with superimposed transparent items in the hull layer. First, transparency is better perceived if there is a good continuity, that is the reason why hulls are drawn with solid contours. Second, transparency is not good to quantitative represent more than five overlapped elements [45], so it becomes only a qualitative guide on complex visualizations, backed up by the piechart layer (see the examples in section 12.5 and chapter 16). The user decides, by means of interaction, which layers to show or hide in order to clarify the visualization.

12.4 INTERACTION

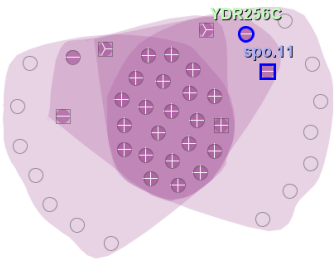
We have implemented several interaction options in order to boost the exploration of group relationships and minimize ambiguities. First, a miniature copy of the display is used to navigate through it by dragging the mouse inside. We have chosen this method instead of the possibly most spread one of tools like [58], that allows the navigation by dragging the mouse in the background of the display, because hulls cover great part of the background, and dragging the background is used for selection of groups of nodes. In addition, this miniature display gives an overview of the complete graph. Elements and biclusters can



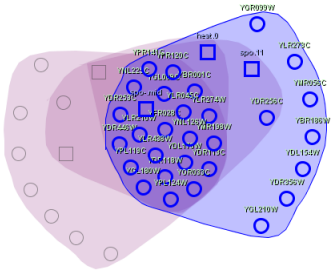
(a) A node hovered over



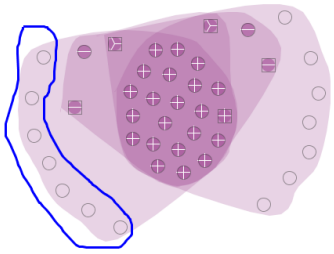
(b) A bicluster hovered over



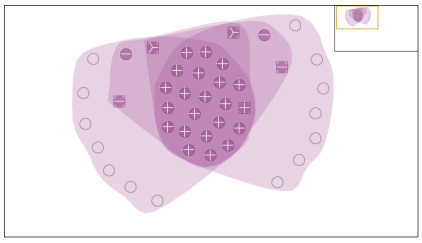
(c) Two nodes selected



(d) A bicluster selected



(e) Selection of several nodes



(f) Overview for navigation

Figure 50: Examples of interaction with Overlapper.

be hovered over and selected (see fig. 50). When an element is hovered over, itself and all their neighbor elements (elements grouped with the hovered element in at least one bicluster) are highlighted in a bright color, facilitating group tracing and reducing the ambiguity that may cause node-zone misplacement. In addition, the name of the hovered element is displayed. When a region is hovered over, all the biclusters intersected in such region are highlighted. If an element is selected it is marked with an identifiable color, keeping its textual label. If a bicluster

is selected, itself and all the elements in it are selected. A right click on an element displays the available information about it (for example, in the case of a gene element: definition, location, GO annotations, etc.). We have implemented several other interaction options further than hovering, selection and navigation; for a detailed review of interaction options see the user guide available with the tool.

12.5 REPRESENTATION OF DIFFERENT RESULT SETS

Color is reserved to convey result sets, so biclusters from different sources could be distinguished and compared in a single visualization. The color of hulls and piechart sectors depends on the result set, as illustrated by fig. 51 with a very simple example. Thanks to hull transparency, the intersection areas of biclusters from different result sets show mixed colors. However, it is hard to quantify how many biclusters intersect in a zone by means of color hue. This is the main reason to use piecharts in order to convey the number of biclusters that an element pertains to (instead of, for example labels with numbers). Color sectors identify the result sets, and at the same time convey the number of biclusters of each type in which the element is in. We have limited the number of different result sets to be displayed together to three to avoid falling into color cluttering. Although, for example, Ware ([141], pp. 125–127) proposes 6 to 12 different colors that can be used to represent classes, we limited it to three colors because the intersecting zones will combine them, thus generating additional colors[6].

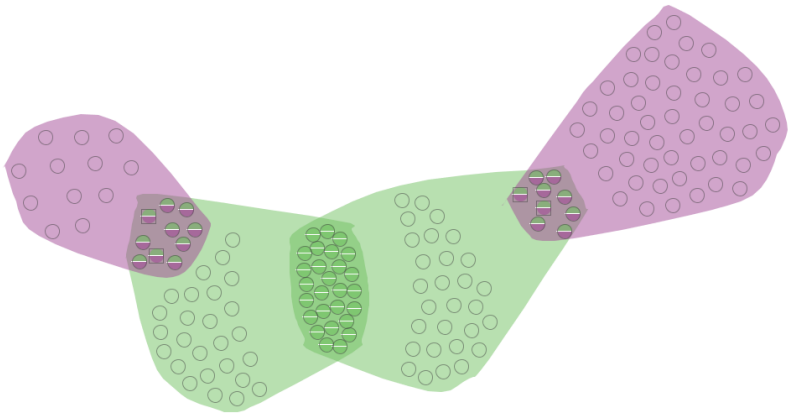


Figure 51: Four biclusters, generated by two biclustering algorithms. Elements in two biclusters, one of each algorithm, have a piechart with two sectors, one of each color.

6 Also note that these colors adds up to the colors selected for hovering and selection

12.6 OVERLAPPER AND VISUALIZATION PRINCIPLES

Overlapper is designed following the rules of information visualization[7], specially the Gestalt laws for group perception [30] and the visualization mantra for the visualization process [120]. Table 8 resumes the main principles observed in Overlapper.

OBJECTIVE	OVERLAPPER	PRINCIPLE
Represent elements	nodes	proximity, shape
Represent biclusters	hulls	continuity, closure
Represent superbiclusters	dual nodes, piecharts	proximity, similarity
Represent result sets	color	color hue
Represent intersections	transparency piecharts	color saturation density
Distinguish biclusters and elements	hulls and nodes	relative size
Dissolve ambiguities	hovering piecharts	color hue shape, density
Overall display	layers, overview	overview first
Reduce cluttering and navigation	layers, interaction	focus+context
Textual information	text layers	details on demand

Table 8: Overlapper visualization objectives and Infovis principles.

7 See section 4.2

13

VISUAL ANALYSIS OF GENE EXPRESSION THROUGH BICLUSTERING

The previous chapter presented a novel visualization technique to represent biclusters that successfully conveys overlap, represents both groups and elements, and helps in the exploratory analysis of groups, supergroups, etc. However, any visualization technique normally focused on just a phase of the analytical process, in this case, the analysis of biclustering results. In addition, it is common that visualization techniques, in order to focus on some aspects of data, disregard other points of view. For example, underground maps easily convey the destinations of trains but disregard geographical positions. In the case of Overlapper, it forfeits expression levels in order to simplify the visualization.

In order to support an analytical process, the visual approach must support all or at least a large part of the analytical points of view and phases. In this chapter we identify the required steps on the gene expression analytical process and describe our approach to support it by means of a visual framework based on multiple-linked views.

13.1 GENE EXPRESSION AND THE ANALYTICAL PROCESS

In order to define a visual analytics approach, Keim et al. [76] identify four major entities: data, visualization, hypothesis and insight. In this section we identified the relevant data and hypothesis for the study of gene expression. Following sections will cover the visualizations selected as relevant for this kind of study and build up a gene expression analysis model based on Keim et al. generic model.

From a *data-centered point of view*, the process for analysis of gene expression starts with input data, that are analyzed generating new information, that it is then compared with available data collections. The relationships are also in the inverse direction: external data are often used to limit or supervise the output of analysis, and analyzed data are also compared with input data in order to validate or interpret them. Therefore, we can divide tasks depending on the kind of data they produce or deal with (see fig. 52):

- *Input data*: designing, building and normalizing transcription data is part of the research area that produces the matrix for gene expression analysis. Input data are inspected in order to find errors during the building process, to check individual gene or condition profiles or to confirm the nature of groups (up or down regulated, constant or coherent profiles, etc.).

- *Analysis data*: several analysis tasks extract relevant information from input data by organizing them into groups, filtering irrelevant data, etc. Data from analysis can be entities by themselves (for example, biclusters or order ranks), but they are usually highly connected to input data, either by reorganization, classification or selection. The analyzed data are inspected in order to confirm or reveal relationships in the input data, explain relation-

ships by means of external data or study relationships among the analyzed groups or classes.

- *External data*: any data relevant to our experiment is worth of inclusion in the analytical process in order to confirm hypothesis or to help explaining our results. External data may come from several sources, from gene ontologies to protein networks.

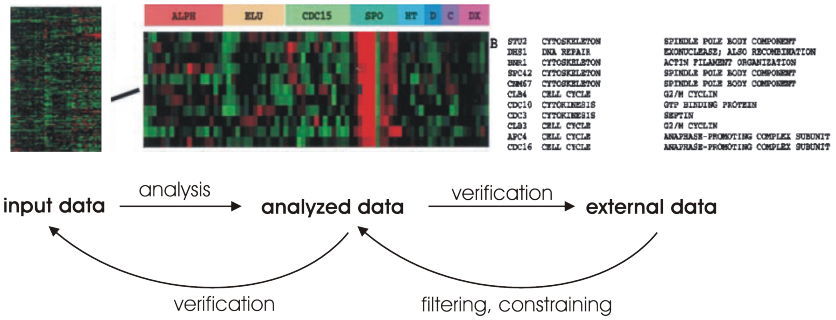


Figure 52: A simple gene expression analysis process from a data-centered point of view. In this example, the genes and conditions of the raw gene expression matrix (left) are reordered and separated following a hierarchical clustering (center, one of the gene clusters has the conditions reordered) and labeled with available information, either from the experiment (for conditions) or from gene ontologies (right). Top image reproduced from [44].

The tasks related to the three kinds of data described above are sequential, starting with input data that are analyzed and then compared with external data, but as in any other analytical process, it is also inherently iterative: external data are used to refine or even design methods that provide new analysis data, confirmation with external data leads to new experiments that generate additional input data, etc.

From a *goal-centered point of view*, we recall Brazma et al. [20] enumeration of the three main questions to be answered by gene expression analysis[1]:

- *Search for expression change along conditions*: inspection of gene profiles and differential analysis are two examples of analysis of expression change. This task is usually more related to hypothesis testing and biomedical applications, with instances such as *does the patient have a high gene expression on cancer-related genes?*

- *Search for expression regulation and gene relationships*: it usually involves data mining methods that infer groups of genes and/or conditions. These grouping tasks are also related to the other two main questions, because they make use of expression change to infer relationships and of existing biological roles to confirm them. This task is more related to systems biology and exploratory analysis, with instances such as *which genes are related to gene rpoH?*

1 The exact formulation of these questions can be found in section 1.1

- *Search for the biological role of genes*: it usually means a conclusion drawn by means of the other two tasks plus the experience of the analysts and the inspection of available sources of knowledge. It is, in the end, a task derived from the study of expression change and expression relationships. A typical instance is *which genes are involved in breast cancer?*

To some extent, these three main tasks are related to the three main kinds of data. Expression change can be inspected on input data, gene relationships makes use of analyzed data and the conclusions about biological roles need from available external data.

Our visual analytics approach must provide the tools to load, generate or retrieve input, analyzed and external data; and to visualize them, within a highly interactive framework that helps with the identified tasks. In order to do that, we think that a multiple-linked views schema is necessary.

13.2 BICOVERLAPPER: A VISUAL ANALYTICS APPROACH

BicOverlapper is the result of our effort to apply visual analytics and information visualization to gene expression analysis and biclustering. It was born as a tool centered in biclustering representation but then it was naturally expanded in the two directions pointed above, dealing with input data and external data. Today, BicOverlapper is a tool that allows the representation of microarray, biclustering and biological data; the biclustering analysis of microarrays and the retrieval of biological data.

In this section we cover the visualization techniques implemented in BicOverlapper and the different kinds of data we can manipulate. Next chapter describes several applications of BicOverlapper in practical cases. This section focuses on the design of the visualizations and its interaction. Usage and technical details about BicOverlapper are available in the user and developer guides, respectively, at `http://vis.usal.es/bicoverlapper`.

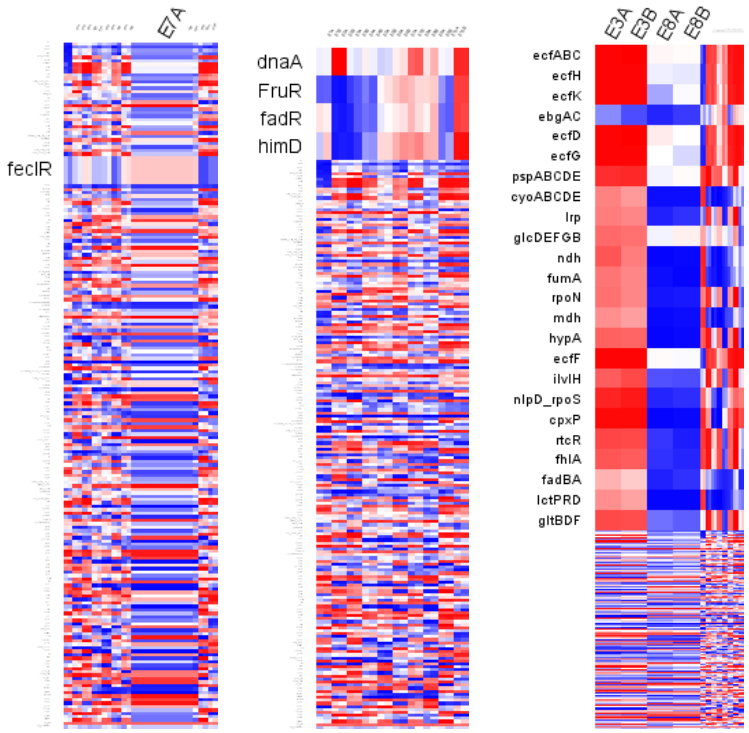
13.2.1 *Visualization Techniques*

BicOverlapper implements several visualization techniques to represent the three identified types of data: input data in the form of gene expression matrices, analyzed data in the form of biclusters and external data in the form of transcription regulatory networks and gene information (essentially, GO terms). Note that the Overlapper visualization technique is not covered in this section, as it is fully described in section 12.1.

Heatmap

The heatmap visualization is a must-be in any visual analytics approach to gene expression analysis. Analysts are used to it and it directly conveys the idea of microarray: color intensities on arrayed spots. About this visualization, we focused on two concepts. The first one is the application of focus+context: heatmap implementations usually rely on reordering and simple zooming in order to inspect the visualization [114, 115, 9]. Although some authors implement more sophisticated zooming or distortion techniques, they partially lose context [104] or require of a previous clustering of data [101].

We implemented a simple bifocal distortion zoom that amplifies the selected gene or condition profiles without losing the context of the rest of the gene expression matrix (see fig. 53). We chose a simple bifocal distortion rather than a more complex distortion, such as fisheye distortion [50, 109], because nearby profiles are not necessarily more important than far-off profiles. This bifocal distortion is also applied to the genes and conditions in a bicluster if it is selected.



(a) (b) (c)

Figure 53: a) Heatmap visualization of a 200x20 expression matrix, with bifocal distortion of gene *fecIR* and condition *E7A*. The color scale goes from bright red (up-regulation) to bright blue (down-regulation). The gene profile shows that *fecIR* is generally low expressed but it is slightly overexpressed on *E7A*. b) Selection of 4 genes, all of them highly overexpressed under the last two conditions. c) Heatmap for a 24×4, additive coherent bicluster, genes and conditions in the bicluster are magnified.

The second concept we wanted to focus on is the color scale. As discussed in section 7.1.1, the typical green-black-red scale is not suitable to perceive changes in hue, so we implemented a blue-white-red scale, where it is easier to distinguish these changes; but allowing the user to switch to a more familiar color scale.

Finally, in order to improve time performance and to save screen real estate, gene expression matrices are not fully displayed. The overall aspect of the data can be perceived by just a sample of the original matrix, and the real utility of a heatmap comes with the filtering and reordering of rows and columns based on analysis results. We set up

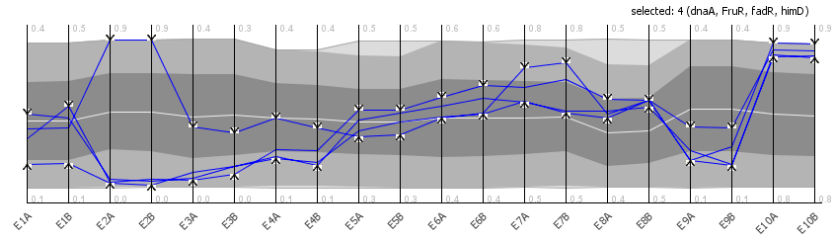
200 rows as enough to display large groups of genes, which can change with the selections performed along the analysis.

Parallel Coordinates

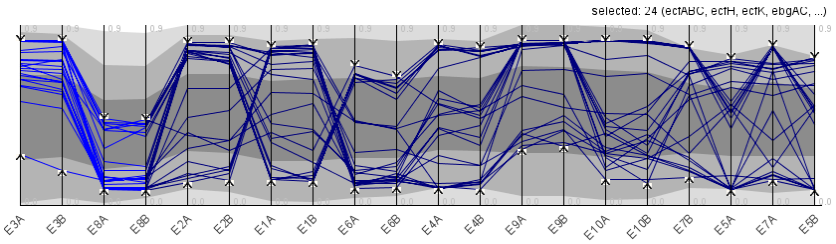
Parallel coordinates are the other main visualization technique used for gene expression. The use of several polylines, one per gene profile, leads rapidly to cluttering. Therefore, as in other tools [114], we substituted them for polygons that convey the overall pattern of gene profiles. Specifically, we computed the mean for each condition, and draw polygons to represent twofold, threefold and fourfold variations around the mean. In addition, lines joining the minimum and maximum expression levels for each condition are drawn.

To select genes depending on their expression levels, we implemented vertical threshold handles for each condition, that can be modified by the user. If 200 or more genes fulfill the threshold criteria determined by the scrolls, a polygon is drawn to avoid polyline cluttering. If they are less than 200, the corresponding polylines are drawn (see fig. 54a).

In the case of biclusters, the coordinates corresponding to conditions that pertain to the bicluster are ordered together at the left, and the segment of the polylines corresponding to those conditions is drawn in a brighter color (see fig. 54b).



(a)



(b)

Figure 54: a) Parallel coordinates for the same matrix represented in fig. 53. The same four genes in fig. 53b are selected. It is clear that one of them diverges in behavior under conditions *E4* and *E3*, and with a mirror effect under *E2* conditions. b) Representation of the bicluster in fig. 53c. The expression profiles of genes in the bicluster, are reordered so the portion corresponding to conditions in the bicluster is drawn first (at the left, in bright blue). The rest of the profile (dark blue) show patterns under conditions not in the bicluster, easier to detect than in the heatmap.

In this implementation we put, as in the case of heatmaps, special care to keep the focus+context philosophy: highlighting selected items

without losing the overall context. Polygons always represent the overall context of genes, and polylines are completely drawn even for conditions outside of the bicluster, in a darker hue. This way, the user can easily determine if the genes in the bicluster are high or low expressed and if there are additional patterns on conditions not in the bicluster.

Bubblemap

Bubblemap is an ancillary visualization technique for bicluster representation. It is implemented for completion and comparison with Overlapper. It is also a convenient summary of biclusters that occupies small screen space and convey overall trends on biclusters.

Bubblemap treats each bicluster as two multidimensional points p_g and p_c, of dimensions n and m (total number of genes and conditions in the expression matrix), respectively. The coordinate i of point p_g is one if gene i is in the bicluster, and zero otherwise (it is analogous for point p_c and conditions). Finally, p_g is projected to one dimension (x) and p_c to another one (y) and a bubble is drawn at that point. The hue, transparency and size of the bubble represent the biclustering method, the internal variation and the size of the bicluster, respectively (see fig. 55). This visualization technique is inspired in mountain maps (see section 7.2), but simplified to 2D in order to reduce 3D-occlusion and computation time.

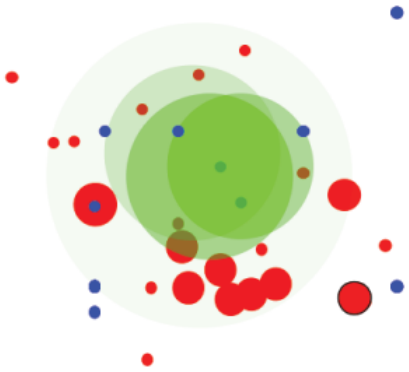


Figure 55: Bubblemap for the results of three biclustering algorithms (on green, blue and red). The red biclustering finds several, small biclusters with low internal variation (solid colors). The green biclustering returns few biclusters, but very large and with high internal variation. Blue biclustering finds very small biclusters. Note that the overlap among bubbles does not necessarily corresponds to the actual overlap among biclusters.

The main drawback of multidimensional projection is the disregarding of details, which in the case of biclusters lead to convey incorrectly overlaps, usually displaying biclusters as much more separated than they actually are (see section 16.1 for an example of it).

Word Cloud

Word clouds are the model example of a vernacular visualization, a technique born outside of the academic world [140]. Its purpose is to

represent the number of occurrences of words in a given text. Our implementation adapts this concept to the field of gene annotations and statistical significance.

Each gene g_i has a corresponding description d_i and a list of annotated GO terms go_i. For a group of genes G, with descriptions D and ontology terms GO, we can extract the single words in every $d_i \in D$, count the number of times they appear in D, and represent them in a word cloud (see fig. 56a). We also extended the concept to di-words (combinations of two words). In a similar way, for ontology terms, again we can determine the number of times that each term appears in GO and represent them in a word cloud (see figs. 56c and d). In addition, we can split the terms in order to show the frequency of single words or di-words. Furthermore, we also offer the option of performing statistical significance tests (in the case of our implementation, hypergeometric tests[2]) in order to know the actual relevance of each GO term in the group respect to the whole set of genes present in the microarray platform, and assign sizes to the terms according to their p-values (see fig. 56d).

(a) (b) (c) (d)

Figure 56: Word clouds for the 20 most highly expressed genes on the human brain according to the microarray data available from [81]. a) Words in gene definitions, number of occurrences in parenthesis. b) MF GO terms. c) BP GO terms. d) BP GO terms, after a hypergeometric test, the bigger terms are the ones with lower p-values (in parenthesis).

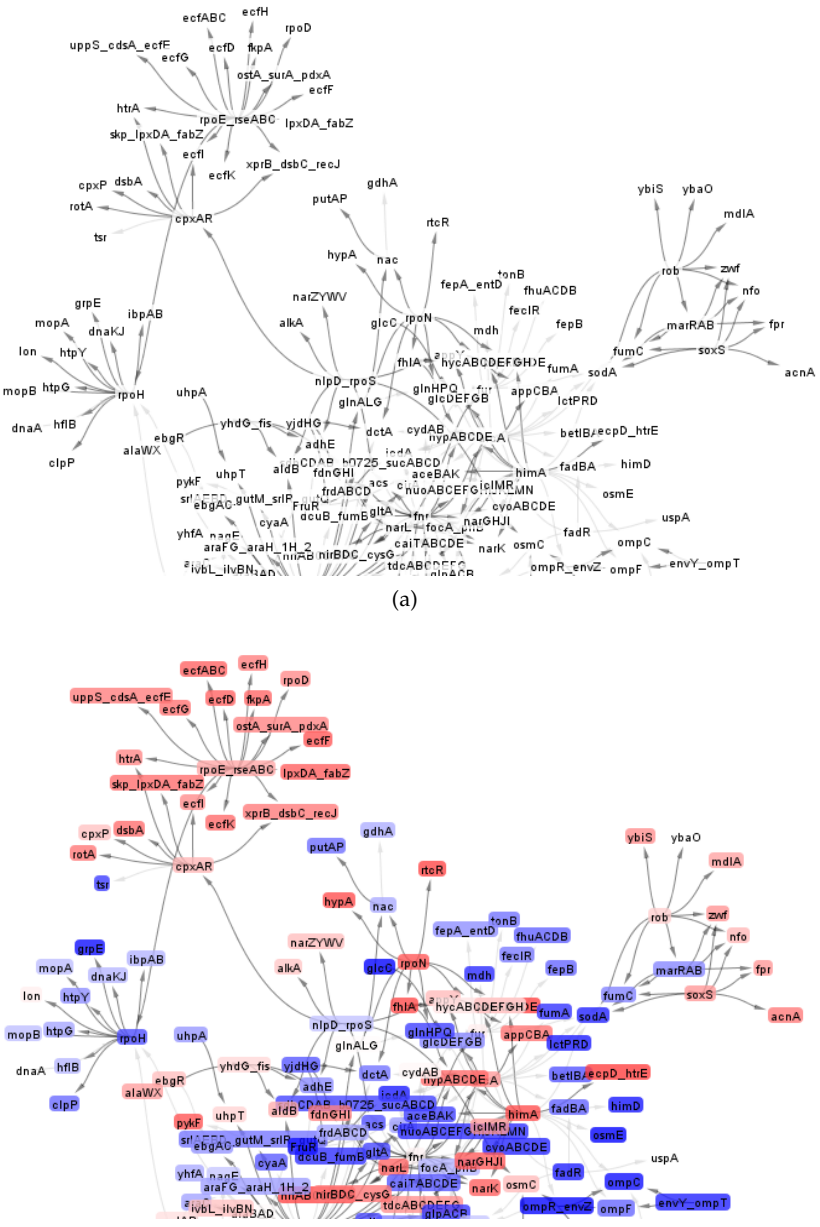
TRN Graph

A Transcription Regulatory Network (TRN) represents transcription relationships among genes. Its usual visualization as graphs represent genes as nodes, and an edge from node *a* to *b* means that gene *a* transcriptionally regulates (activates or inhibits) gene *b*. There are several tools that visualize TRNs, such as Cytoscape [116] or Osprey [21] (see [96] for a survey.), with different display models. We have selected a typical force directed layout model [48] to display the network (see fig. 57), with labels for nodes and curved lines[3] for edges, directed from

2 We use the implementation in BioConductor package *GOstats*

3 Curved lines are better than straight lines because they partially reduce edge cluttering. The improvement is specially important in nodes that have several nodes coming out/in, because curved lines tend to bundle together when they are close to them.

the regulator gene to the regulated one (dark grey for activation and light grey for inhibition). Nodes can be colored with expression levels using the same scale as the heatmap if a single condition is selected.



(a)

(b)

Figure 57: a) Detail of a synthetic TRN generated by SynTReN [36] for the same example used in figs. 53, 54. The degree of complexity that TRNs may reach is evident, but at this scale (a few hundred genes) it is possible to detect regulators at the center of the bundles of edges, such as *rpoH* or *rpoE_rseABC* (top and left). b) The same figure colored by the expression levels under the condition *E3A*.

We designed this visualization as simple as possible to avoid cluttering. However, TRNs grow everyday[4], and the display of thousand of nodes with an even larger number of edges becomes a problem for visualization that, as of today, does not have an easy solution.

13.2.2 *Data Communication and Retrieval*

Following the multiple-linked views philosophy, the visualization techniques implemented by BicOverlapper are interconnected so the interaction with one of them affects the rest. A communication layer translates the items selected on a visualization to the related items on other visualizations (see fig. 58). Despite their nature, all the data sources share two entities, genes and conditions, that are used to perform the translations. For example, the selection of a gene on the heatmap leads to the selection of its GO terms on the word cloud or the biclusters it pertains to in the Overlapper. Details about how selection affects to linked visualizations can be found in the user guide [5].

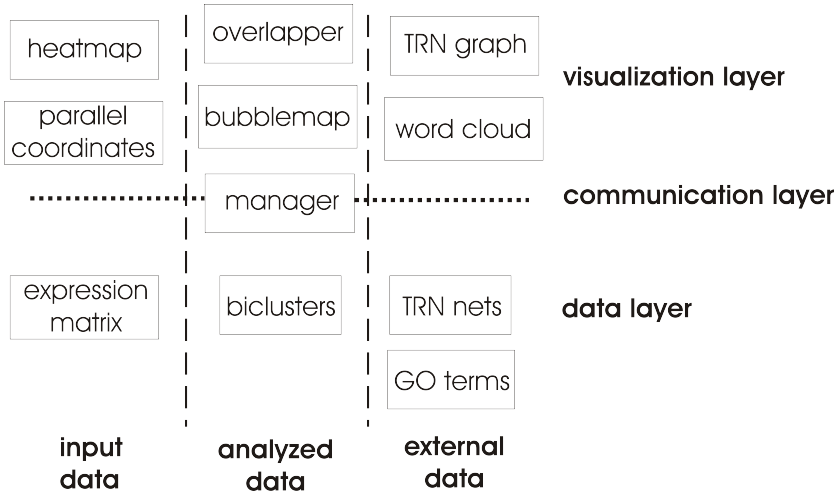


Figure 58: Layer and data schema of BicOverlapper.

Another interesting aspect to discuss is how data of different nature are retrieved by BicOverlapper. BicOverlapper treats data retrieval in three major ways:

- *User load*: the straightforward method to obtain data is to let the user to provide them. We tried to minimize this use because it usually require changes in data formats that may be tedious and produce errors. However, it is important to give the user control about the nature of data, specially in the case of expression data (the starting point of an analysis), but also in the case of analyzed data (the number of possibilities of analysis is so large that it is impossible to cover them all in a single tool). Therefore, users must manually load any expression matrix they want to

4 For example, the TRN for *Escherichia coli* increased from 577 relationships in 2002 [118] to 2724 in 2004 [82]

5 Available at `http://vis.usal.es/bicoverlapper`

analyze. The format is kept simple so the user's data can be easily translated from other formats. However, it could be interesting to allow automatic retrieval of microarrays from the public repositories[6]. On the other side, although BicOverlapper can run some biclustering algorithms, we also allow the load of biclusters from other sources. In this case, we have kept BicAT format for biclusters, because it is simple enough for our needs and because BicAT comprises several additional biclustering algorithms[7]. Finally, TRNs can also be manually loaded with a GraphML format[8]. Detailed information about data formats is available in the user guide.

- *Biclustering analysis:* bicluster results can be generated by BicOverlapper, by means of interfaces to the biclustering algorithms implemented in the package *biclust*. Sometimes, bicluster results generate a huge number of biclusters, or very large biclusters, so we included the option of bicluster post-filtering, using the method described in [99].

- *Automatic retrieval:* external data are usually available on the Internet. By now, BicOverlapper retrieves genetic information from NCBI Entrez Gene[9], QuickGO[10] and BioConductor annotation packages. It is important that the microarray data contains proper gene identifiers in order to avoid mismatches with gene retrieval interfaces.

13.2.3 *Data Interaction*

All the visualizations implements several method to interact with them. The choosing of proper interaction options is key in order to facilitate the analytical discourse. It is important to implement familiar interaction techniques so the user feel familiar with them, keeping the dialogue with the tool intuitive. At the same time, the interface must provide elements to develop a discourse. For example, if we defined that the study of the expression of single genes is relevant, a text search for gene names should be available. Details about how these interactions are carried out are fully described in the user guide, but following there is a summary of the main interaction techniques implemented:

- *Selection and hovering*: the visual items representing genes, conditions and biclusters can be selected or hovered over. Hovering has the effect of highlighting the element and displaying additional information, such as labels. Selection usually has a similar effect, but it is permanent until the element is unselected or another element is selected. In addition, the selection is sent to other views in order to highlight the corresponding visual items in all of them. Multiple elements can be also selected, and we use different color hues to distinguish hovered items from selected items.

6 ArrayExpress, for example, has recently released a BioConductor package to retrieve microarray experiments that could be integrated in BicOverlapper
7 Note that our format for biclusters is flexible enough to load any kind of groups, not only biclusters, providing that they comprise genes, conditions or both
8 http://graphml.graphdrawing.org
9 http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene
10 http://www.ebi.ac.uk/QuickGO

- *Zooming and panning*: in order to navigate through them, an intuitive zooming and panning technique is provided for heatmaps, TRN graphs and bubble maps.
- *Text searches*: the search for genes is usually demanded by users. Gene names can be searched in TRN graphs and in the biclustering visualizations. In biclustering visualizations both gene and condition names may be searched, highlighting them (in the Overlapper) or the biclusters they pertain to (in the bubblemap).
- *Other interactions*: each visualization has special characteristics that give way to particular interactions. These kind of interactions are described on their corresponding sections. They include, for example, the distortion of profiles in heatmaps, the selection of profiles by expression thresholds in parallel coordinates, or the selection of the splitting performed on descriptions in word clouds.

13.2.4 *BicOverlapper and the Analytical Process*

Along this chapter we have identified the major data, goals and visual techniques involved in a visual analysis of gene expression and how BicOverlapper integrates them. This can be translated to the visual analytics process defined by Keim et al. [76] and discussed in section 5.1 (see fig. 59).

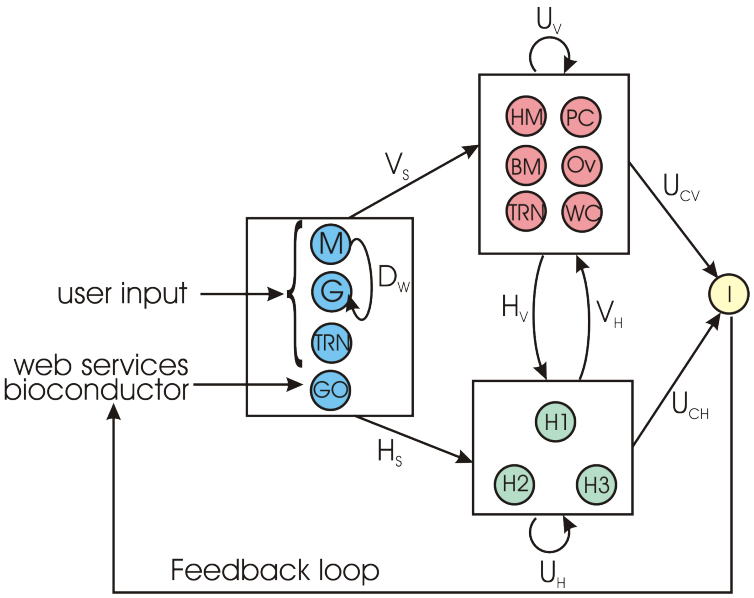


Figure 59: Keim et al. [76] adapted schema to gene expression analysis with BicOverlapper.

There are four types of gene-related data:

- The gene expression matrix (M), provided by the user.
- The groups (G), either introduced by the user or generated by the tool (biclusters) by the D_W process.

- TRNs, also provided by the user.
- GO terms, retrieved from different resources.

These data are represented via V_S with 6 visualizations:

- Heatmaps (HM) and parallel coordinates (PC) visualize M and, by means of reordering and filtering, they also partially visualize G.
- Bubblemaps (BM) and Overlapper (Ov) visualize the whole set of groups G.
- TRN graphs visualize TRNs
- Word clouds (WC) visualize GO terms.

Several interactions (U_V) are implemented to modify the visualizations, as described in section 13.2.3.

Besides, we have three main types of hypothesis, which are instances of the three main questions discussed in section 13.1:

- Expression variations (H1). Its instances are more related to M, HM (bifocal distortion of profiles) and PC.
- Gene relationships (H2). H2 instances are more related to G, BM and specially Ov, and supported by the links to HM and PC.
- Biological roles (H3). H3 instances are related to external resources (TRN, GO) and their visualizations (TRN, WC).

The relationship between hypotheses and visualizations is in both directions. H_V represents the process by which a given visualization raises or confirms a hypothesis. V_H represents the process by which a hypothesis leads to build or search for new visualizations. The following section describes an example of how BicOverlapper answer to some questions of the three types.

Finally, insight (I) is produced from the confirmation or refusing of hypothesis (U_{CH}) but also directly from the inspection of the visualizations (U_{CV}). Insight represents the knowledge distilled from the testing of hypothesis and the inspection of visualizations

Part VI

RESULTS

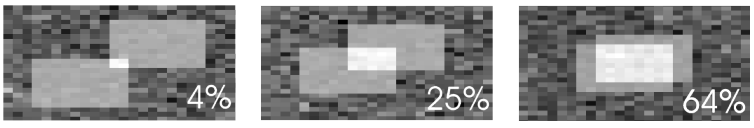
14 APPLICATION OF EXTERNAL AND RELATIVE INDICES

In chapter 11 we adapted the $\bar{\Gamma}'$ statistic to biclustering. Now, we are going to test its capacity for determining the matching of biclustering results to the data matrix. In order to do this, we design an experiment that will compare the performance of our $\bar{\Gamma}'$ internal measure with an external measure (the F1 measure) in the task of determining the best parameter settings for two biclustering algorithms. The implementation of the biclustering algorithms in order to perform these and other experiments gave way to the implementation of a R package for biclustering that is also briefly described in this chapter.

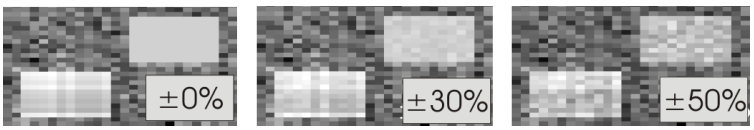
14.1 PARAMETRIZATION OF BICLUSTERING ALGORITHMS

We want to find the best parameter configuration for two biclustering algorithms under two different datasets. The search for the best parameter setting is made by means of two measures, the F1 measure and the $\bar{\Gamma}'$ measure.

Regarding data, we build two sets of synthetic 100×50 matrices with embedded biclusters in a similar way to the previously discussed validations (see section 6.6). The first set of matrices contains two constant 10×10 biclusters with overlapping degrees from 0% to 100%, with 10% increments[1] (see fig. 60a). The second set has two non-overlapping biclusters, one constant and the other one additive coherent, with normal distribution random noise. The distribution deviation increases from 0 (no noise) to 1, with 0.1 increments (see fig. 60b). In both sets, the expression levels out of the embedded biclusters are just random noise.



(a) Overlap in biclusters



(b) Noise in biclusters

Figure 60: Details of some of the synthetic matrices for the procedure.

The algorithms selected for the test are Bimax [99] and the improved Plaid Model of Turner et al. [135] (from now on, just the Turner algorithm). Bimax is one of the most compared biclustering methods, by means of non-biological and biological validation, usually yielding

1 The overlap degree is the same in both rows and columns

a high rank [99, 103, 90]. The Turner algorithm was tested by their authors with several data sets.

Both methods were implemented in R according to the specifications in the corresponding bibliography (see section 14.2 below). Each algorithm has been tested with several parameter configurations (all the combinations of parameters in table 9) for each overlap/noise level. The parameter ranges try to cover parameter intervals around the values proposed by the authors, excluding ranges very permissive or restrictive that would lead to several biclusters or no biclusters at all, respectively. For example, for the Turner algorithm the range proposed by its authors for both t_1 and t_2 is $[0.5 - 0.7]$, but the parameters may be in $[0 - 1]$. However, values very close to 0 or 1 return no biclusters, so we have slightly enlarged the range to $[0.4 - 0.8]$.

Bimax		Turner	
PARAMETER	RANGE (STEP)	PARAMETER	RANGE (STEP)
Min. rows	3–9 (1)	t_1	0.4–0.8 (0.1)
Min. columns	3–9 (1)	t_2	0.4–0.8 (0.1)
Binary threshold	1–10% (1)		

Table 9: Selected ranges for biclustering parameters.

The procedure for the experiment is to run the algorithm presented in section 11.2 (fig. 44) for each noise or overlap degree, depending on the data set. The experiment is done for both algorithms, Bimax and Turner Plaid Model. Fig. 61 shows the results of such procedure.

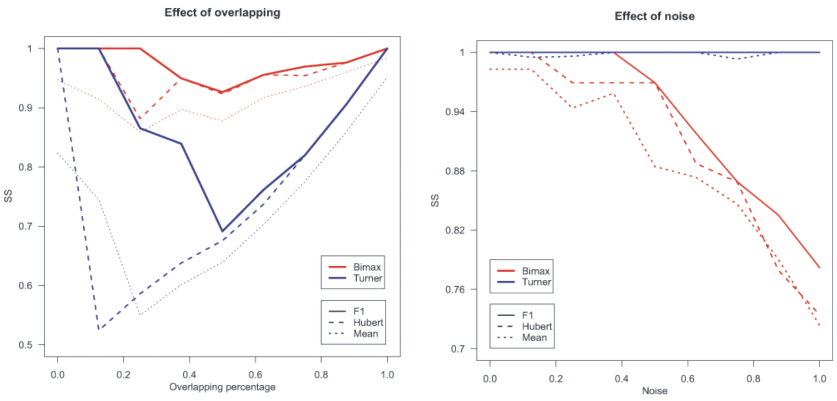


Figure 61: Best SS measure achieved by using the F_1 and the adapted Hubert ($\bar{\Gamma}'$) statistics, along with the mean of SS for all the tested configurations, for different levels of overlap (left) and noise (right). The study is carried out for two biclustering algorithms, Bimax (red) and the Turner Plaid Model (blue).

Regarding the biclustering algorithms, we compare the solid lines that refer to their best parameter setting. When the noise or overlap level is

low, Bimax (in red) finds the exact embedded biclusters without adding spurious biclusters (SS = 1). The performance downgrades slightly in biclusters overlapped around a 50%, but it increases again with high overlaps (the biclusters are almost the same). However, performance downgrades more in the case of noise, because some of the biclusters' expression levels are too low due to noise, and therefore are disregarded during the binarization step of Bimax. The Turner algorithm (blue) is unaffected by these levels of noise, but it is very sensitive to overlap. This is because the pruning phase (included in this algorithm to improve the original plaid model algorithm) fails when trying to prune the overlapped parts of the biclusters.

Regarding the validation indices, F_1 (continuous lines) gives always the best possible result, because it makes use of *a priori* information about the actual embedded biclusters. Choosing the best configuration according to $\bar{\Gamma}'$ (dashed lines) gives suboptimal results that sometimes coincide with the best result, and almost always improve the average solution (point lines).

We think that this study reveals the importance of choosing the correct configurations of a biclustering algorithm in order to perform a comparison. Following these results, choosing a wrong configuration can downgrade the performance of an algorithm up to 50% in terms of SS. The use of F_1 for external validations is clear, but to use the proposed adapted Hubert statistic in real case applications, where a priori knowledge is not available, could also improve the performance. The drawback of computing relative indices is the time performance, because the algorithm must be run once per configuration[2]. For small synthetic matrices, it is not so critical because, as stated in [99] the running-times are never above 120 seconds on a present-day personal computer[3], but it could be a problem on real datasets, specially for biclustering algorithms with several parameters.

14.2 IMPLEMENTATION OF BICLUSTERING ALGORITHMS IN R

In order to apply the proposed method for biclustering parametrization, and several other tests and analyses, the R package *biclust* [72] has been developed in collaboration with Sebastian Kaiser and Friedrich Leisch from the University of Munich. The R programming language [65, 35] is oriented to its use in statistics, but in the last years its use in bioinformatics has dramatically increased thanks to BioConductor [52], an open software for computational biology and bioinformatics.

Biclust implements five biclustering algorithms covering different search techniques and bicluster types (see chapter 6). The implemented algorithms are Bimax [99], xMotifs [87], Cheng and Church biclustering algorithm [34], Turner et al. Plaid Model [134, 135] and Spectral biclustering [77]. In addition, the package includes basic representations of biclusters in heatmaps, parallel coordinates and bubble maps. It also includes some statistical metrics such as the Jaccard index and measures of variance; and some pre-processing methods, such as discretization and binarization.

biclust *is available as part of the CRAN project (*`http://cran.r-project.org`*)*

2 In the case of the Bimax configurations considered in table 9, the algorithm must be run 490 times

3 For example, the computer used for these tests has a 2.8 GHz CPU with 2GB of RAM

15 APPLICATIONS OF OVERLAPPER

With the application of Overlapper to the analysis of biclustering results on gene expression, the user can have an overview of the biclustering results, and then follow an exploratory approach to search for relationships, superbiclusters, etc. In this chapter we briefly discuss three applications of Overlapper to real cases, although the biological analysis will be enhanced with the visual analytics approach to be discussed in the next chapter. The last section presents other applications of Overlapper to the visualization of groups in areas unrelated with gene expression and biclustering.

15.1 APPLICATION TO A CONTROLLED REAL CASE

The first example is a controlled real case. By controlled be mean that we have restricted the biclustering algorithm to fit the known characteristics of the data. We analyzed Chen et al. [31] gene expression study for *Schizosaccharomyces pombe* under five environmental stress conditions, tested at two times (15 and 60 minutes after exposition to stress) and compared against non-stress conditions (zero minutes after exposition to stress). We applied the Bimax algorithm in order to search for groups of genes with specific up-regulation under the stress conditions. The expression matrix is binarized with a threshold of 4, so only transcription levels at least 4 times higher than the corresponding transcription level for the control condition (no stress) are considered. Bimax returns several biclusters, but we filtered the ones with two conditions, corresponding to the two time frames of a stress condition, thus obtaining five groups related with specific stresses[1]. These five groups are then visualized with Overlapper (see fig. 62).

Just with an overview, several conclusions can be drawn from the visualization:

1. The five biclusters present a high degree of overlap. A central group of genes are overexpressed under any stress condition, and some more are expressed in four or three conditions (conveying $S_{3,4,5}$ superbiclusters).
2. Oxidative stress, heat shock and heavy metal stress are the conditions that provoke overexpression in more genes. They have a large number of genes shared two by two for these conditions (S_2 superbiclusters).
3. There are no genes highly overexpressed uniquely under DNA damage (labeled MMS), and just two under osmotic stress (labeled Sb). Most of their genes respond to several stress conditions.
4. Most of the conditions for oxidative stress (labeled H_2O_2), heavy metal stress (labeled Cd) and osmotic stress have been also biclustered for the DNA damage bicluster (condition nodes with two sector piecharts).

1 Chen et al. called these groups SESR

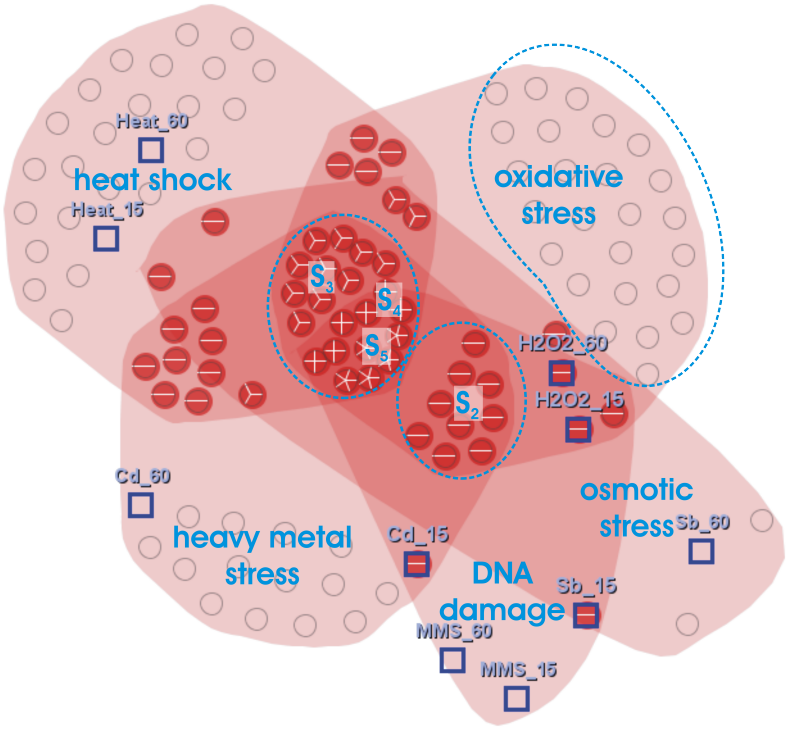


Figure 62: Stress biclusters visualized with Overlapper.

Even considering that our method to search for stress groups is different from the one in [31][2], some of their statements are quickly confirmed with our visualization. For example, oxidative stress presents a large overlap with heavy metal and heat stresses, and the genes related to DNA damage are almost completely included in the group of genes related to oxidative stress[3].

The static visualization in fig. 62 reveals a drawback intrinsic to set diagrams that was discussed in previous sections: it is impossible to show every group with perfectly defined zones if the number of elements, groups and the degree of overlap is moderately high. In this case, it occurs with the superbicluster of order 2 surrounded by the dashed line (S_2). This group of genes is in the heavy metal and oxidative stress groups, but not in the DNA damage or osmotic stress groups. The two-sector piecharts and, especially, the interaction with the tool, such as hovering over the nodes in the superbicluster, dismisses the ambiguity.

2 The method of Chen et al. is based on a differential analysis of gene expression under each kind of condition, and it determines a gene as condition-related if it is over-expressed in just one of the two times measured for each condition (it also must not be highly over-expressed under other conditions)

3 In our example, all of them are included, while they identify two exclusive DNA damage-related genes

15.2 APPLICATION TO A NON-CONTROLLED REAL CASE

The second example is a non-controlled real case, meaning by non-controlled that we have not restricted the biclustering algorithm parameters to fit any characteristic of the data. The Bimax algorithm is applied to Eisen et al. [44] yeast expression data, resulting in a very large set of biclusters. Overlapper reveals a high overlap among biclusters, forming a large group at the top of the visualization, and a smaller one at the bottom (see fig. 63). Genes in the top group are mainly biclustered by sporulation conditions (*spo.5*, *spo.7*, *spo.mid*, etc.) and the ones in the bottom group are biclustered by heat shock conditions (*heat.20*, *heat.40*).

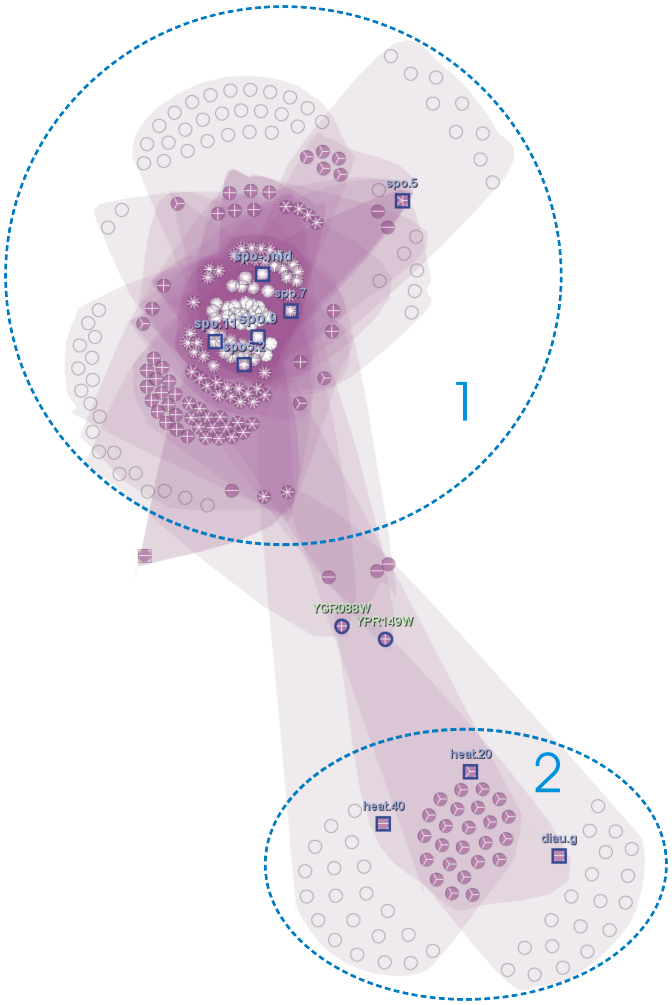


Figure 63: 50 biggest Bimax biclusters found on Eisen et al. yeast expression data [44]. Overlapper reveals two main bicluster groups, a large one formed by genes grouped by sporulation conditions such as *spo.5* and *spo.7* (1); a smaller one is formed by genes mainly grouped by heat shock conditions *heat.20* and *heat.40* (2). Some genes, specially *YGR088W* and *YPR149W* are in biclusters of both groups (we call them "bridge genes").

Sporulation is a reproductive method that is adapted for dispersal and surviving for extended periods of time in unfavorable conditions.

Heat shock stress occurs when the organism is exposed to high temperatures. A possible biological interpretation of the results could conclude that sporulation and heat shock are the conditions under which genes are very highly over-expressed. Heat shock is an unfavorable condition, so it seems reasonable that it is related to sporulation. The gene *NCE102* (*YPR149W*), highly expressed under both conditions, is related with membrane, secretion and protein transport and secretion, probably related with the thicken of the cellular wall in spores, also a defensive mechanism against extreme heat. We call these genes related with two different functional groups *bridge* genes. Other bridge gene, *CTT1* (*YGR088W*), is related to stress conditions, specially heat and oxidative damage, but relationship to sporulation cannot be found in their gene annotations, which could be subject of further experiments. Using a visual analytics approach for this exploratory process would improve the utility of Overlapper. How that has been carried out in the present work is covered in the next chapter.

15.3 VISUAL COMPARISON OF BICLUSTERING ALGORITHMS

One last example is illustrated in fig. 64. Here, two biclustering algorithms have been applied to a synthetic dataset for *E. coli*[4]. Turner et al. [135] plaid model (in red), implements an additive coherent model, while Spectral Biclustering [77], in purple, implements a multiplicative coherent model.

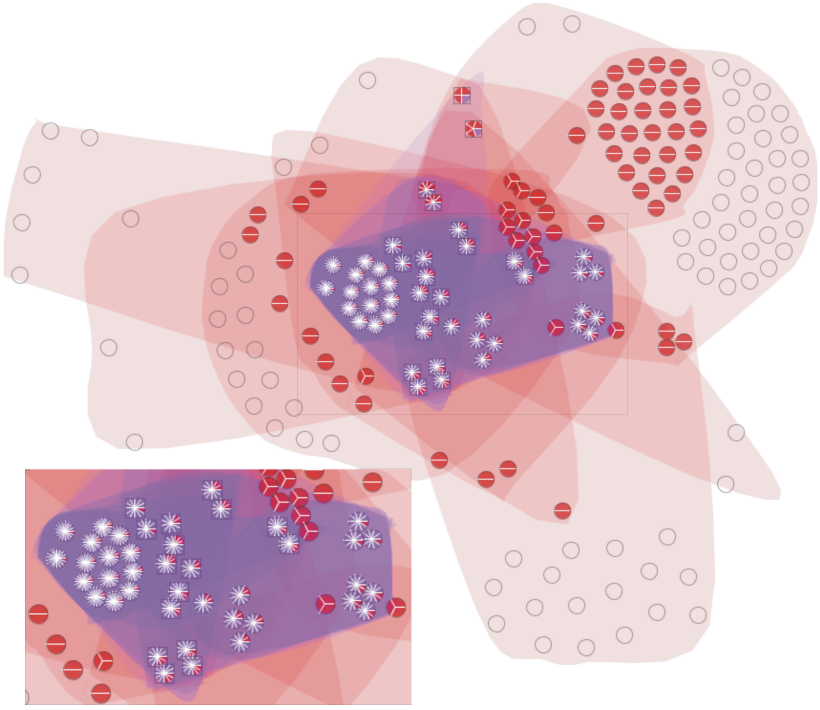


Figure 64: Spectral (purple) and Turner (red) biclusters for *E. coli* synthetic data.

4 This dataset will be further discussed in section 16.2

Overlapper shows that Spectral biclusters are highly redundant, while Turner biclusters are more spread. The inspection of piecharts reveals that the elements grouped by Spectral biclustering are included at least in a Turner bicluster (i. e. there are no completely purple piecharts), but the opposite is not true. A possible conclusion is that the data do not follow a multiplicative model but an additive model, because the only biclusters that Spectral biclustering finds are mainly included in Turner biclusters, that is: they are groups with very low multiplicative factors that can be captured with an additive model. This is confirmed by the additive model used to build the synthetic gene expression matrix (see [36], addendum).

15.4 OTHER APPLICATIONS OF OVERLAPPER

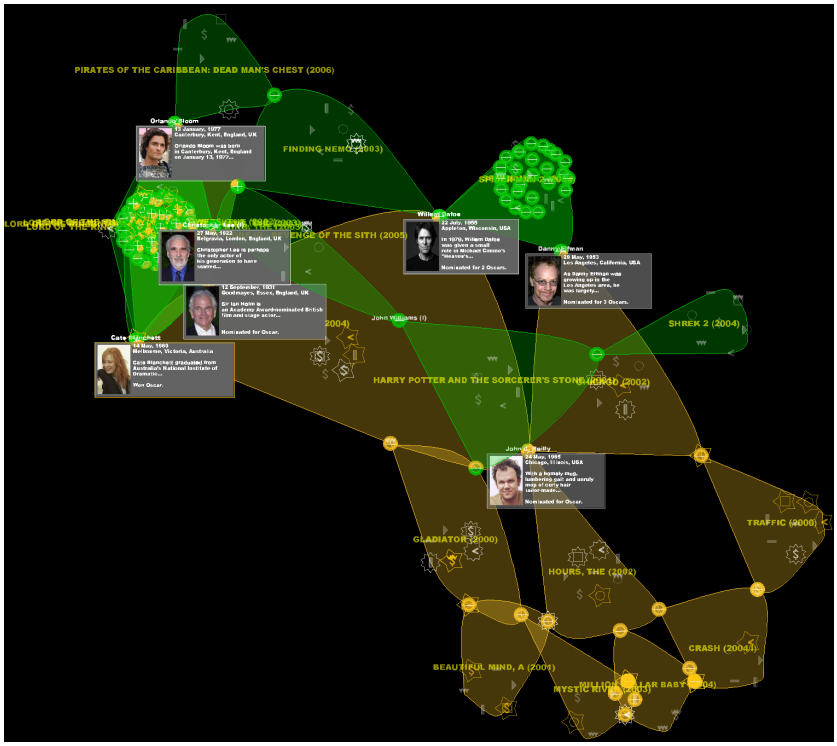
The representation of groups and group relationships is useful for almost any field, because groups are intrinsic to a large number of data sets. Information about movies, scientific papers or terrorism, for example, share two levels of data, the individuals related to the field (actors, researchers or terrorists) and their relationships (movies, papers, organizations). Usually, these collaborations overlap, having individuals in more than one group.

In addition, groups can be inferred in almost any data set. That is the case of data clustering, which usually searches for non-overlapping groups. This is also the case of complex queries in databases. For example, searches can be done for data that fulfill queries A, B and C, and then a comparison of the relationships among the three result sets may be interesting. Furthermore, some grouping algorithms consider overlapping as essential for searching groups, such as biclustering algorithms.

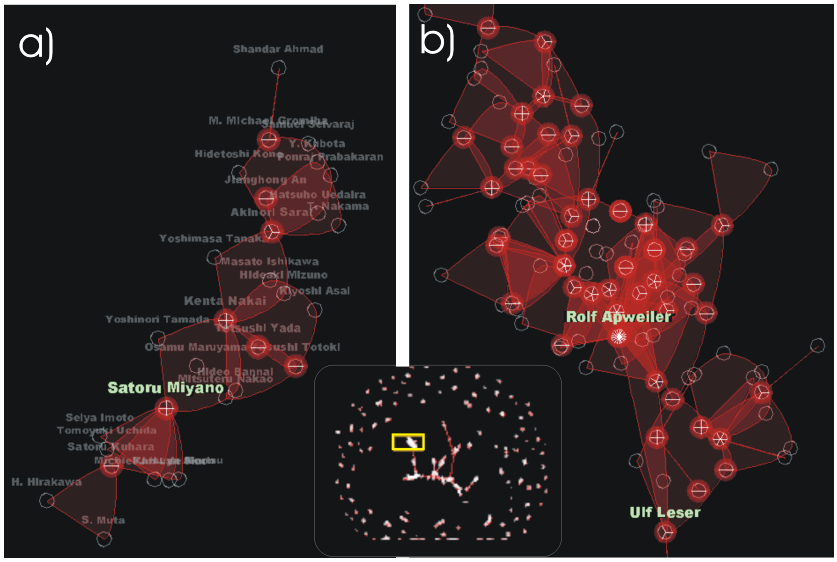
We have applied the Overlapper visualization technique to several areas (see figs. 65, 66):

- *Film relationships*: in order to participate in the Infovis'07 contest, Overlapper was adapted to visualize relationships among actors, directors, writers, etc. (elements) stored in the movies they made (groups). Our contribution was selected among the finalists for the contest [129]. Additionally, it won the Graph Drawing'07 contest [41].
- *Co-authorship networks*: the same idea can be applied to co-author networks, using authors as elements and papers as groups [105]. The results facilitate to infer research groups from publications, and to determine their collaborations and key authors.
- *Music tags*: tag clouds can have structure, with tags included in larger tags, such as in the case of music genres[5]. In [32], we designed an evaluation study based on task completion which shows an improvement of the representation of hierarchical structure in tag clouds with Overlapper, compared to traditional tag clouds representation.
- *Idea organization*: finally, we have also applied our visualization technique to some typical cases found in literature, such as KJ diagrams for idea organizing [92].

5 For example, rock has several subgenres such as hard rock or pop rock. Pop rock subgenre is also in the pop genre



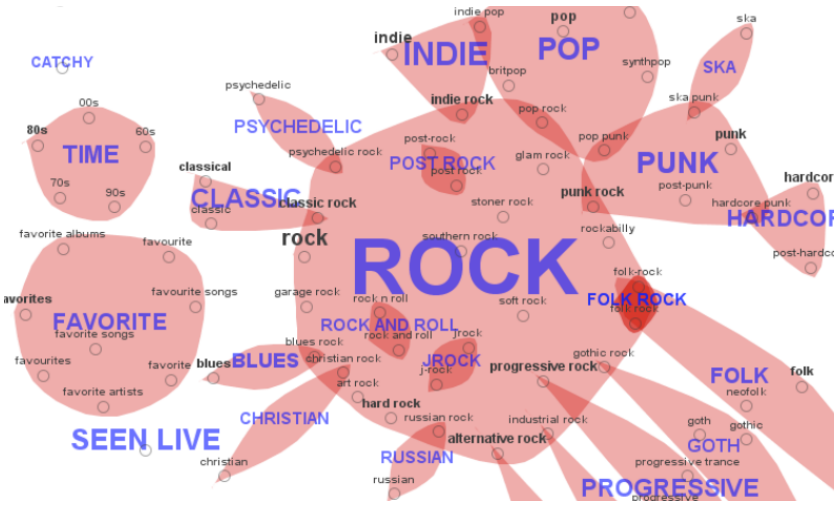
(a) Film relationships



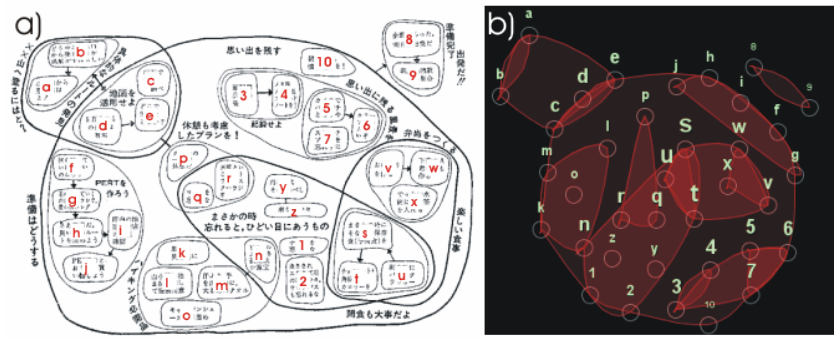
(b) Co-authorship networks

Figure 65: a) Top ten most awarded (yellow) and most profitable (green) movies between 2000 and 2006. Details of key individuals highlighted. b) Two supergroups of researchers according to papers in the journal *Bioinformatics*. Overview of the full representation at the center-bottom.

All of these applications have natural groups intrinsic to data, so its validation is easier than the studied case of biclusters. Also, most of them have a low number of elements per group, which makes the final display with overlapper very clear. The feedback from these applications was vital to refine our visualization design and to plan future lines of research.



(a) Music tags



(b) Idea organization

Figure 66: a) Adaptation of Overlapper to represent tag clouds as groups (image found in [32]). b) Example of a complicated structure reproduced from [92] (left) and its representation with Overlapper (right).

16 VISUAL ANALYSIS SUPPORTED BY BICOVERLAPPER

We discuss in this chapter several applications of BicOverlapper for different gene expression datasets, showing how the proposed visual analytics approach helps to discover useful information about gene expression. Biological explanations are given and confirmed when possible (for example, if synthetic datasets are used or we can find similar results in already published papers). Other possible biological explanations are given as examples of how we can retrieve relevant information by means of a proper visual analysis, but they will require a deeper experimental study in order to confirm them, which is out of the scope of this document.

16.1 S. POMBE MICROARRAY EXPERIMENT

This example analyzes Chen et al. [31] microarray experiment for the study of environmental stress in *Schizosaccharomyces pombe*[1]. This experiment comprises transcriptional profiling for 5 different stress conditions, each one checked two times, 15 and 60 minutes after the application of stress, and compared against the non-stress condition (right before the application of stress). The main focus of the experiment is to determine which genes are involved in every stress condition (CESR group) and which ones are involved in just a specific stress condition (SESR groups). Basically, this is a mixture of H1 (change in gene expression, for example from high expression in a specific stress condition to normal expression in the rest) and H3 (biological functions related to the genes involved in stress). We describe here some visual analysis procedures to answer these kind of questions with BicOverlapper.

A first approach to answer these questions is the use of PC to perform a visual clustering of genes. For example, if we fit the threshold handles to select up-regulation under conditions related to H_2O_2 and down-regulation for the rest, we capture 14 genes[2] which are in the SESR group for oxidative stress (see fig. 67). We can complement it with WC to give an idea of the biological processes and molecular functions involved with this group. The GO terms highlighted by WC confirm the conclusions of Chen et al. [31], such as the relationship to *oxidation reduction*, to ion-related processes or to *pyridoxine (pyridoxal phosphate binding)*. This whole process is an example of how the V_H flux works: driven by H1 question (*which genes are specifically overexpressed for* H_2O_2) we inspect the visualizations of PC and HM with the corresponding filters, generating the visualization of WC with the corresponding genes.

1 This experiment is available from ArrayExpress with accession name *E-MEXP-29*. The processed data were slightly adapted to fit BicOverlapper's format, basically adding information about the species (*Schizosaccharomyces pombe*) and the source for gene annotation (there is no available BioConductor annotation package for this organism, so gene annotations were retrieved from web services)

2 This group is smaller than the one obtained by Chen et al. for H_2O_2. This is because their method considered a gene as stress-specific if just one of the two time conditions is highly up-regulated, which involves more genes.

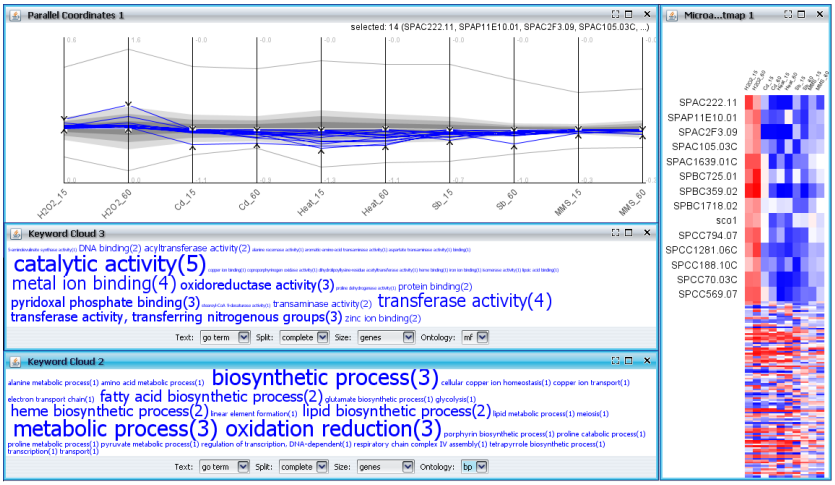


Figure 67: H_2O_2 SESR group obtained by means of PC interaction in BicOverlapper.

The process described above is somehow supervised, because we know what we are searching for (genes specifically up-regulated for H_2O_2 stress). This supervised search can be performed by other methods, such as reproducing Chen et al. method to find SESRs or trying to replicate it with biclustering or other classification method. The result of both approaches is to visually confirm what it is known and published, as discussed in section 15, making conclusions to become evident so they can be drawn quicker, reducing the time required for analysis.

On the other side, we can try to perform a non-supervised analysis.To do this, we can run a biclustering algorithm without a special configuration of parameters or post-processing. For example, if we choose the Bimax algorithm[3] we can find a group of 5 biclusters partially out of the main trend of the rest of biclusters[4] (see fig. 68). All of these 5 biclusters include the *MMS_15* condition, which is not grouped by any other bicluster (it has a five-sector piechart). These five biclusters have 16 genes not grouped by any other biclusters, as it is shown in the visualization. If we select them and visualize the annotated GO terms, 4 out of the 16 genes are annotated with *meiosis*, a term that just appears in one out of the 44 genes in the central group .

The *MMS_15* condition relates with the response to the stress provoked by Methylmethane sulfonate (MMS), an agent that generates DNAdamage in the cell [31]. Meiosis is a cell process in which DNA damage occurs in the form of DNA double strand breaks (DSBs) [64], which possibly points to the function of these genes under this stress condition.

3 We selected a minimum of 10 genes per bicluster, a binary threshold of 4% and filter highly overlapped groups in order to keep the result set on a moderate size (50 biclusters)

4 The main group of biclusters corresponds to genes in CESR

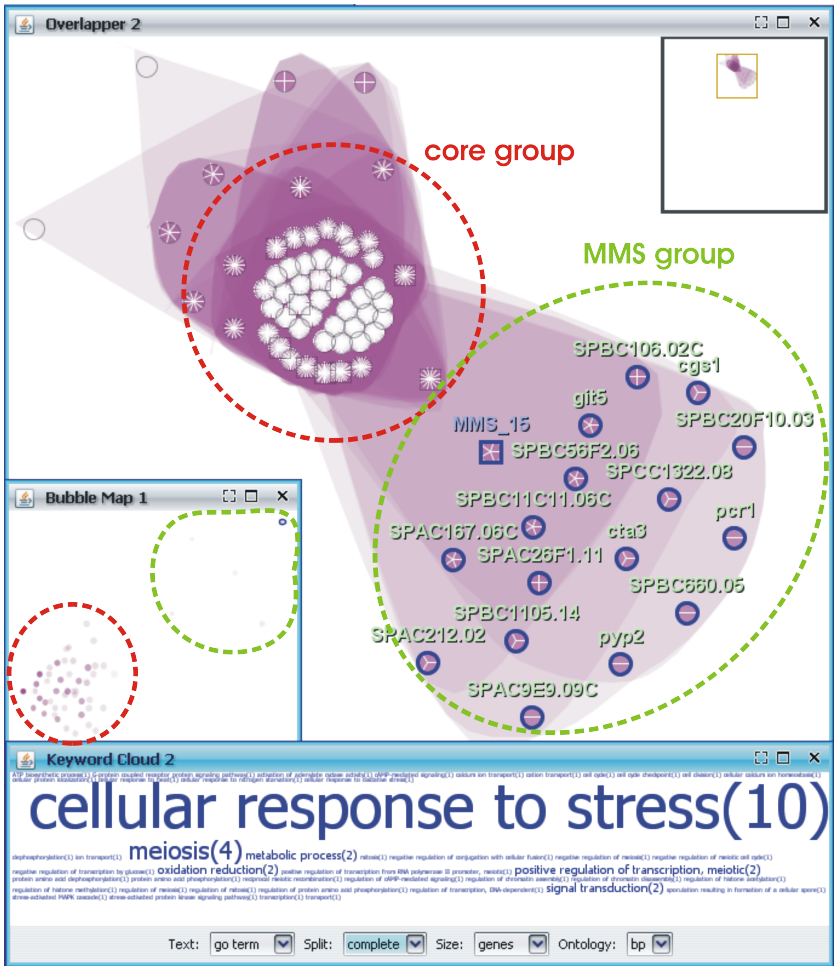


Figure 68: Visualization of Bimax results. Five biclusters are clearly separated from the trend, comprising 16 genes apart from the central group, and condition *MMS_15*. Four of these genes are annotated with meiosis.

16.2 E. COLI SYNTHETIC MICROARRAY EXPERIMENT

In order to provide an example of TRN visualization, we chose a small synthetic 200×20 matrix generated by SynTReN [36]. This software generates expression levels that depend on the connections among genes in a given TRN. We limited it to 200 genes related by the *E. coli* network described by Shen-Orr et al. [118], because larger graphs become too complex to visualize.

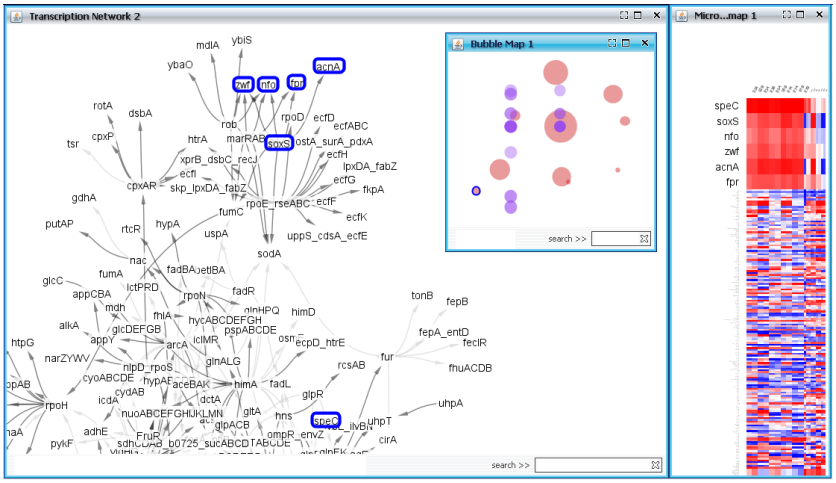
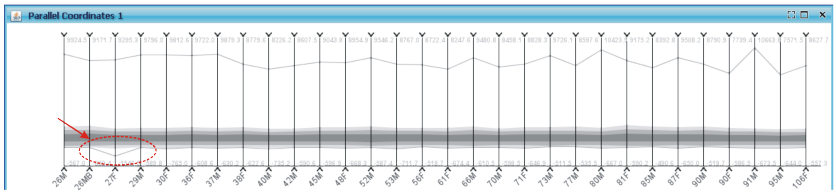


Figure 69: *E. coli* microarray represented in a heatmap, a bubblemap with two biclustering results in different colors, and a TRN graph with the corresponding regulation network. A bicluster is selected in the bubblemap (surrounded in blue at the left bottom), that contains over-expressed genes (highlighted in the heatmap) related to *soxS* regulation, but also a gene not regulated by it, *speC*.

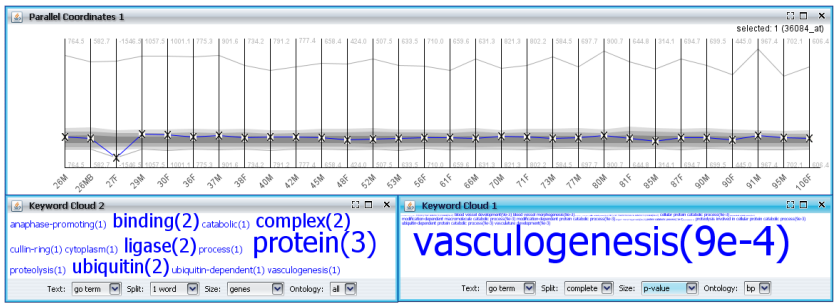
The placement of nodes depending on regulation relationships is useful as a kind of "visual biological clustering" and also it is useful to validate biclusters. If all the genes in a given bicluster are connected in the network, the bicluster is biologically good but do not reveal additional information. If they are disperse on the TRN, the bicluster is probably spurious. It they are into two separate areas, maybe it leads to previously unknown relationships in the network. For example, the bicluster selected in fig. 69 groups six genes, five of them regulated by *soxS*, but one located in a completely different area of the network, *speC*. On a real case, this could imply a direct or indirect regulation of *speC* by *soxS*, under certain conditions (such as the ones grouped by the bicluster).

16.3 HUMAN BRAIN MICROARRAY EXPERIMENT

Finally, this is an example of how visual analysis could arise unexpected questions with the Lu et al. [81] microarray experiment for the study of ageing in the human brain[5]. This example comprises transcriptional profiling for 30 individuals from 26 to 106 years of age[6]. The main task for its authors was to determine which biological processes decay or increase with age. Basically, this is a mixture of H1 (analysis of change in gene expression) and H3 (relationship to biological functions). However, a new question arises with a simple glance to PC: without further interaction, this visualizations shows that, for condition *27F*, the minimum expression level is far away of the fourfold deviation of the mean, which does not happen for other conditions (see fig. 70).



(a)



(b)

Figure 70: a) A deviation in the minimum expression level of *27F* (the third condition starting by the left) can be easily detected. b) Further inspection reveals that the probe *36084_at* (corresponding to gene *CUL7*) is differentially under-expressed respect to the rest of conditions. WC relates it to *ubiquitin*, *ligase* and *vasculogenesis*.

The inspection of the lowest expression levels for condition *27F* reveals that only the expression profile of *CUL7* (probe *36084_at* on the Affymetrix chip) has an under-expression greater than 4-fold for the patient. If we display WC in order to visualize the biological roles of this gene, we see that it is related to *ubiquitin* and *ligase*, but specially to *vasculogenesis*, after a statistical significance test. Further inspection of the condition reveals that the behavior of other genes is also different from samples of similar age, either by using PC or by the inspection of Lu et al. figures. This discovery could lead to further study outside our

5 This experiment is available from ArrayExpress with accession name *E-GEOD-1572*, or from GEO with accession name *GSE1572*. We must download it manually and slightly adapt it to fit BicOverlapper format, which basically is to add information about the species (*Homo sapiens*) and the microarray platform (*hgu95av2* of Affymetrix)

6 We refer to each condition as NS, being N the age and S the sex, for example 42M

tool to identify if this individual had some relevant disorder (the tissue samples of the study were neuropathologically normal for age, but it could be any other disorder relevant for expression in brain tissue) and if this could bias the analysis. This is an example of how the flux H_V, hypothesis generation from visualizations, works (see section 13.2.4).

Part VII

CONCLUSIONS

17 CONCLUSIONS

Reasoning draws a conclusion, but does not make the conclusion certain, unless the mind discovers it by the path of experience. — Roger Bacon

The main outcomes of the research undertaken for this thesis were the development of a new technique for the visualization of biclusters, and its integration into a framework for the visual analysis of gene expression data. This novel visualization technique allows to visualize several biclusters together within a single representation, which conveys precisely the main characteristics of biclustering. The visualization prioritizes gene and conditions over expression levels, allowing to easily explore the relationships among genes and conditions inferred by biclustering algorithms. This visualization technique is part of a framework that provides visualizations for other related data, either expression matrices or external biological knowledge. This whole set of highly interactive visualizations plus the integration of biclustering analysis, provides a complete framework for the visual analysis of gene expression data by means of biclustering.

The benefits of a visual approach to gene expression analysis is well known thanks to several techniques and tools [44, 114, 104, 101]. Some of these approaches already pointed out to principles that are now covered by visual analytics. A first conclusion of this work is that it is possible to formally adapt the analysis of gene expression to visual analysis by identifying the involved data sources, visualization techniques and analysis tasks. To make these identifications gives the possibility of designing a framework that fulfills the needs of the analyst. When tested with controlled cases, our approach is validated because it finds the already known characteristics of data. Furthermore, when tested against non-controlled cases, even with just superficial knowledge about biology, thanks to this approach we were able to uncover gene relationships and other circumstances.

Biclustering is a relatively new area for the analysis of gene expression, and the lack of standards and visualization techniques discourages potential users. On the other side, for example, clustering is very used and has very well defined algorithms (hierarchical clustering and k-means clustering) and visualizations (heatmap and dendrogram, parallel coordinates). After dealing with biclustering algorithms during the development of our research, we think that, in order to achieve the same degree of success with biclustering, it is necessary a process of validation of biclustering methods and the techniques used to visualize them. Biclustering and other advanced analysis methods will be more demanded by the scientific community with the progressive understanding of gene expression and other biological processes, so it is important to define techniques and standards for validation and visualization that support them.

A relevant finding of our research is that the biclustering comparisons usually do not take into account the parameter configuration of biclustering algorithms. We showed that the design of an appropriate internal index and its use in a simple parametrization procedure can

significantly improve the performance of biclustering algorithms, thus allowing to compare the best configurations of biclustering algorithms.

Another relevant conclusion is that biology is such a fertile field that it is very heterogeneous. We found heterogeneity on biclustering methods, on microarray technologies and on biological knowledge. Although the heterogeneity is good in order to provide different points of views and solutions for each problem, this variation should be minimized in the underlying data structures that allow to share the information. Identifiers, file formats and naming conventions should keep an order that back up the "chaos" in the techniques. It is not always easy to separate both aspects, and heterogeneity spreads in, for example, expression file formats and gene identifiers. In addition, this heterogeneity of knowledge also affects researchers. In order to improve the analysis of biological data, there is a need of collaboration between specialists in the design and interpretation of experiments (biologists, doctors, etc.) and the data management and analysis of their outcomes (statisticians, computer scientists, etc.)

With regard to the findings presented in this thesis, we can extract some conclusions about the development of visual analysis approaches for bioinformatics problems. First, it is key to identify and verbalize the questions that we want to answer, and the tasks that will help to answer them. It is also fundamental to identify the data we require in order to answer these questions and to characterize the main entities of these data. Second, the decisions about the design of visualization techniques must take into account the data we have, the questions we want to ask, the available kinds of analyses, and the characteristics of human perception. It is also very important to consider the state of the art representations for the data we want to represent, so we do not reinvent the wheel, and to respect the most accepted approaches so we do not to confuse the user. Regarding interaction, it must be intuitive so the user doesn't feel confused, but at the same time must help in the analytical discourse. The interaction response should be quick enough in order to keep the user in control of the tool. Time-consuming tasks should be optimized, or the user must be properly advised. Finally, data formats should only be redesigned if the available formats do not fulfill our needs to a large extent. The time spent reformatting or correcting formats sometimes exceed the time spent analyzing data. About data generated by external entities and suspected to change with time, such as gene annotations, they must be retrieved from external entities when possible; otherwise we need to be aware of every possible change on these data.

A general conclusion to be drawn from this thesis is that the use of multiple-linked, highly interactive views, designed in order to answer well defined tasks, has an enormous potential to reduce analysis times and uncover relevant information. While the research on the visualization of overlapped groups is still an open area of study, it is key to interpret complex group relationships such as the ones derived from biclustering analysis. Genes may be involved into several functions, collaborating with different genes on each one, so the visualization of overlapping groups is very valuable for the study of functional genomics. We hope that this thesis will guide other developers towards considering the use of visualization techniques of overlapped groups and visual analytics approaches to achieve similar results.

17.1 FURTHER WORK

There are several opportunities for further research based on the study presented in this thesis. First, the *biclust* package can include several other biclustering algorithms that performed well in the surveyed comparisons. In addition, the package can also include the proposed $\bar{\Gamma}'$ index and the method to find the best parameters for biclustering algorithms.

Second, now that we defined and tested an internal validation index for biclusters, we can use it extensively to test its usefulness in real cases with no a priori information, and in biclustering comparisons. Before this, we also want to perform additional tests with several biclustering algorithms that implement different search methods and kinds of biclusters.

Third, BicOverlapper can grow in several directions. One is the development of a more efficient way to visualize TRN networks. A possible approach is to visualize just the selected elements and their nearest neighbors. Other improvement in the networks visualization is the detection of network motifs, that have been used for biclustering comparison by some authors. We are also considering the use of other relevant biological networks, such as metabolic pathways. A second direction is the deeper consideration of the visualization of the information about conditions. The MIAME definition for experiment-related data or the very recent Experimental Factor Ontology (EFO)[1] could be good starting points to visualize experimental factors in word clouds. Another direction under study is the use of misplacement metrics and semantic zoom in order to improve the Overlapper visualization technique. This visualization can become very cluttered with very large biclustering sets, although we think that it is not very useful to visualize very large biclustering result set (more than, say, 100 biclusters) because our experience reveals that these results are very redundant and often inconclusive. Finally, but probably the most important direction, it will be very interesting to perform user tests to measure the usability of the tool. The feedback from such tests will give us new improvement directions.

1 http://www.ebi.ac.uk/efo/index.html

BIBLIOGRAPHY

[1] The human genome project. URL http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml. (Cited on page 3.)

[2] M. D. Adams, S. E. Celniker, R. A. Holt, C. A. Evans, J. D. Gocayne, et al. The genome sequence of Drosophila melanogaster. *Science*, 287(5461):2185–2195, 2000. URL http://www.sciencemag.org/cgi/content/abstract/287/5461/2185. (Cited on page 3.)

[3] J. S. Aguilar-Ruiz. Shifting and scaling patterns from gene expression data. *Bioinformatics*, 21:3840–3845, 2005. URL http://bioinformatics.oxfordjournals.org/cgi/reprint/21/20/3840. (Cited on page 53.)

[4] C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Conference on Human Factors in Software, CHI '94. 1994. (Cited on page 39.)

[5] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000. URL http://www.nature.com/nature/journal/v403/n6769/pdf/403503a0.pdf. (Cited on pages 17, 26, and 64.)

[6] W. G. Alvord, J. A. Roayaei, O. A. Quiñones, and K. T. Schneider. A microarray analysis for differential gene expression in the soybean genome using bioconductor and r. *Briefings in Bioinformatics*, 8(6):415–431, 2007. URL http://bib.oxfordjournals.org/cgi/content/abstract/8/6/415. (Cited on page 4.)

[7] D. R. Anderson, K. P. Burnham, and W. L. Thompson. Null hypothesis testing: problems, prevalence, and an alternative. *Journal of Wildlife Management*, 64(4):912–913, 2000. URL http://www.warnercnr.colostate.edu/~anderson/PDF_files/TESTING.pdf. (Cited on pages 59 and 88.)

[8] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, et al. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, 25:25–29, 2000. (Cited on page 18.)

[9] S. Barkow, S. Bleuer, A. Prelic, P. Zimmermann, and E. Zitzler. Bicat: a biclustering analysis toolbox. *Bioinformatics*, 22(10):1282–1283, 2006. (Cited on pages 55, 57, 67, 69, 71, and 113.)

[10] T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, D. Rudnev, et al. NCBI GEO: mining tens of millions of expression profiles–database and tools update. *Nucl. Acids Res.*, 35:D760–765, 2007. (Cited on page 17.)

[11] W. Basalaj. Proximity visualization of abstract data. Technical Report 509, University of Cambridge Computer Laboratory, 2001. URL http://www.pavis.org/essay/. (Cited on page 69.)

[12] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987. (Cited on page 39.)

[13] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of Computational Biology*, 10:373–384, 2003. (Cited on pages 53, 54, and 60.)

[14] S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Phys. Rev. E*, 67(3):031902, 2003. URL http://link.aps.org/doi/10.1103/PhysRevE.67.031902. (Cited on page 60.)

[15] D. P. Berrar, C. S. Downes, and W. Dubitzky. Multiclass cancer classification using gene expression profiling and probabilistic neural networks. In *Pacific Symposium on Biocomputing*, volume 8, pages 5–16. 2003. (Cited on pages 25 and 58.)

[16] D. P. Berrar, W. Dubitzky, and M. Graznow, editors. *A Practical Approach to Microarray Data Analysis*. Kluwer Academic Publishers, 2003. (Cited on pages 11, 23, 25, and 26.)

[17] F. Bertault and P. Eades. Drawing hypergraphs in the subset standard. In *Graph Drawing: 8th International Symposium*. 2000. (Cited on pages 74, 75, and 103.)

[18] F. R. Blattner, G. Plunkett, C. A. Bloch, N. T. Perna, V. Burland, et al. The complete genome sequence of Escherichia coli K-12. *Science*, 277(5331):1453–1462, 1997. URL http://www.sciencemag.org/cgi/content/full/277/5331/1453?ijkey=YhbCy4Zd0/4mw. (Cited on page 3.)

[19] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, et al. Minimum information about a microarray experiment (MIAME)–toward standards for microarray data. *Nat Genet*, 29(4):365–371, 2001. (Cited on page 20.)

[20] A. Brazma, J. Vilo, and E. G. Cesareni. Gene expression data analysis. *FEBS Lett*, 480:17–24, 2000. (Cited on pages 4 and 112.)

[21] B.-J. Breitkreutz, C. Stark, and M. Tyers. Osprey: a network visualization system. *Genome Biology*, 4(3), 2003. (Cited on page 117.)

[22] R. Bringhurst. *The Elements of Typographic Style*. Version 2.5. Hartley & Marks, Publishers, Point Roberts, WA, USA, 2002. (Cited on page 163.)

[23] D. Brodbeck and L. Girardin. Design study: using multiple coordinated views to analyze geo-referenced high-dimensional datasets. In *Proceedings. of the International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 104–111. 2003. (Cited on page 38.)

[24] T. Buering, J. Gerken, and H. Reiterer. User interaction with scatterplots on small screens - a comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):829–836, 2006. URL http://hci.uni-konstanz.de/~buering/pdfs/infovis2006.pdf. (Cited on page 39.)

[25] A. Buja, D. Cook, and D. Swayne. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5:78–99, 1996. (Cited on page 40.)

[26] S. Busygin, G. Jacobsen, and E. Kramer. Double conjugated clustering applied to leukemia microarray data. In *Second SIAM ICDM, Workshop on clustering high dimensional data*. 2002. (Cited on pages 53 and 54.)

[27] A. Califano, G. Stolovitzky, and Y. Tu. Analysis of gene expression microarrays for phenotype classification. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 8:75–85, 2000. (Cited on page 54.)

[28] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Information Visualization: Using Vision to Think*. The Morgan Kaufmann Series in Interactive Technologies. 1999. (Cited on pages 30 and 31.)

[29] P. Carmona-Saez, R. D. Pascual-Marqui, F. Tirado, J. M. Carazo, and A. Pascual-Montano. Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC Bioinformatics*, 7(78), 2006. URL www.biomedcentral.com/1471-2105/7/78. (Cited on pages 26, 53, 58, 64, and 89.)

[30] D. Chang, L. Dooley, and J. E. Tuovinen. Gestalt theory in visual screen design. In *Seventh world conference on computers in education*. 2002. (Cited on pages 36 and 110.)

[31] D. Chen, W. Toone, J. Mata, R. Lyne, G. Burns, et al. Global transcriptional responses of fission yeast to environmental stress. *Mol. Biol. Cell*, 14:214–229, 2003. URL http://www.sanger.ac.uk/PostGenomics/S_pombe/docs/214.pdf. (Cited on pages 17, 23, 26, 64, 129, 130, 137, and 138.)

[32] Y.-X. Chen, R. Santamaria, A. Butz, and R. Theron. Tagclusters: Semantic aggregation of collaborative tags beyond tagclouds. In *Lecture Notes in Computer Science. Smart Graphics 2009*. 2009. (Cited on pages 133 and 135.)

[33] K. O. Cheng, N. F. Law, W. C. Siu, and T. H. Lau. Bivisu: Software tool for bicluster detection and visualization. *Bioinformatics*, 2007. (Cited on pages 55, 57, 67, and 71.)

[34] Y. Cheng and G. M. Church. Biclustering of expression data. *Proc. Int'l Conf Intell Syst Mol Biol.*, 8:93–103, 2000. (Cited on pages 53, 54, 60, 66, and 127.)

[35] M. J. Crawley. *The R Book*. 2007. (Cited on page 127.)

[36] T. V. den Bulcke, K. V. Leemput, B. Naudts, P. van Remortel, H. Ma, et al. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7(43), 2006. URL http://www.biomedcentral.com/1471-2105/7/43. (Cited on pages 118, 133, and 140.)

[37] H. Doleisch and H. Hauser. Smooth brushing for focus+context visualization of simulation data in 3d. *Journal of WSCG*, 10(1):147–154, 2002. URL http://www.vrvis.at/via/research/smooth-brush/. (Cited on page 40.)

[38] S. Dudoit, J. Fridlyand, and T. Speed. Comparison of discrimination methods for the classification of tumors usin gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002. (Cited on page 26.)

[39] S. Dudoit, Y. H. Yang, M. J. Callow, and T. P. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, 12:111–139, 2002. (Cited on page 26.)

[40] D. Duffy and A. Quiroz. A permutation based algorithm for block clustering. *J. Classification*, 8:65–91, 1991. (Cited on page 54.)

[41] C. A. Duncan, S. G. Kobourov, and G. Sander. Graph drawing contest report. Technical Report, 2007. (Cited on page 133.)

[42] P. Eades and Q.-W. Feng. Multilevel visualization of clustered graphs. In *Proc. Graph Drawing, GD*, 1190, pages 101–112. Springer-Verlag, Berlin, Germany, 1996. URL `citeseer.ist.psu.edu/eades97multilevel.html`. (Cited on page 75.)

[43] R. Edgar, M. Domrachev, and A. E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acid Res.*, 30(1):207–210, 2002. (Cited on page 17.)

[44] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95:14863–14868, 1998. (Cited on pages 4, 26, 64, 68, 92, 112, 131, and 145.)

[45] J.-D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *IEEE Symposium on Information Visualization*. 2002. (Cited on pages 35 and 107.)

[46] J. Flower and J. Howse. Generating Euler diagrams. In *Diagrammatic Representation and Inference: Second International Conference*. 2002. URL `http://cmis.mis.brighton.ac.uk/Research/vmg/papers/D2K2FH.pdf`. (Cited on page 75.)

[47] J. Flower, P. Rodgers, and P. Mutton. Layout metrics for euler diagrams. In *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*, pages 272–280. 2003. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1217990`. (Cited on pages 75 and 106.)

[48] T. M. J. Fruchterman and E. M. Reinhold. Graph drawing by force-directed placement. *Software – Practice and Experience*, 21:1129–1164, 1991. (Cited on pages 104 and 117.)

[49] B. Fry. *Computational Information Design*. Ph.D. thesis, MIT, 2004. URL `http://acg.media.mit.edu/people/fry/phd/`. (Cited on pages 44, 45, and 79.)

[50] G. W. Furnas. Generalized fisheye views. In *Human Factors in Computing Systems*, pages 16–23. 1986. URL `http://www.si.umich.edu/~furnas/`. (Cited on page 114.)

[51] N. Gehlenborg, J. Dietzsch, and K. Nieselt. A framework for visualization of microarray data and integrated meta information. In *Information Visualization*, 4, pages 164–175. Pallgrave Macmillan Ltd., 2005. (Cited on pages 64 and 65.)

[52] R. Gentleman, V. Carey, D. Bates, B. Bolstad, M. Dettling, et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80, 2004. URL http://genomebiology.com/2004/5/10/R80. (Cited on page 127.)

[53] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc. Natural Academy of Sciences US*, 97(22):12079–12084, 2000. (Cited on page 54.)

[54] G. A. Grothaus, A. Mufti, and T. Murali. Automatic layout and visualization of biclusters. *Algorithms for Molecular Biology*, 1(15), 2006. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?&pubmedid=16952321. (Cited on pages 69, 70, 71, and 89.)

[55] M. Halkidi, Y. Batisfakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2/3):107–145, 2001. URL http://www.springerlink.com/content/k43h06u025w2x4q6/fulltext.pdf. (Cited on page 58.)

[56] J. A. Hartigan. Direct clustering of a data matrix. *J. Am. Statistical Assoc.*, 67(337):123–129, 1972. (Cited on page 54.)

[57] J. Heer and D. Boyd. Vizster: visualizing online social networks. In *IEEE Symposium on Information Visualization*. 2005. URL http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/&toc=comp/proceedings/infovis/2005/2790/00/2790toc.xml&DOI=10.1109/INFOVIS.2005.39. (Cited on page 76.)

[58] J. Heer, S. K. Card, and J. A. Landay. Prefuse: a toolkit for interactive information visualization. In *Proceedings of SIGCHI Human Factors in Computing Systems*, pages 421–430. ACM Press, New York, NY, USA, 2005. URL http://jheer.org/publications/. (Cited on pages 39, 76, and 108.)

[59] Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000. (Cited on pages 104 and 105.)

[60] M. A. Hibbs, N. C. Dirksen, K. Li, and O. G. Troyanskaya. Visualization methods for statistical analysis of microarray clusters. *BMC Bioinformatics*, 6(115), 2005. URL http://www.biomedcentral.com/content/pdf/1471-2105-6-115.pdf. (Cited on pages 64 and 65.)

[61] I. Hovatta, K. Kimppa, A. Lehmussola, T. Pasanen, J. Saarela, et al. *DNA Microarray Data Analysis*. CSC - Scientific Computing Ltd., 2005. URL http://www.csc.fi/csc/julkaisut/oppaat/arraybook_overview. (Cited on pages 11 and 23.)

[62] J. Howse, G. Stapleton, J. Flower, and J. Taylor. Corresponding regions in euler diagrams. In *Diagrammatic Representation and Inference: Second International Conference*. 2002. URL http://www.cmis.

brighton.ac.uk/Research/vmg/papers/D2K2HSFT.pdf. (Cited on page 73.)

[63] R. Hubbard. Why we don't really know what "statistical significance" means: a mayor educational failure. *Journal of Marketing Education*, 28:114–120, 2006. URL http://jmd.sagepub.com/cgi/content/abstract/28/2/114. (Cited on pages 59 and 88.)

[64] N. Hunter. Hop1 and the meiotic DNA-damage response. *Cell*, 132:731–732, 2008. (Cited on page 138.)

[65] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996. (Cited on pages 40 and 127.)

[66] K. Ikeo, J. Ishi-i, T. Tamura, T. Gojobori, and Y. Tateno. Cibex: center for information biology gene expression database. *C. R. Biologies*, 326(10-11):1079–1082, 2003. URL http://www.cib.nig.ac.jp/gfr/research_pdf/CIBEX.pdf. (Cited on page 17.)

[67] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985. (Cited on page 66.)

[68] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *First IEEE Visualization Conference*, pages 361–381. IEEE Computer Society, 1990. (Cited on page 39.)

[69] International Human Genome Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001. URL http://www.nature.com/nature/journal/v409/n6822/pdf/409860a0.pdf. (Cited on page 3.)

[70] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988. (Cited on pages 23, 27, 58, 97, and 98.)

[71] D. Kahneman, A. Treisman, and B. J. Gibbs. The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, 14:175–219, 1992. (Cited on page 34.)

[72] S. Kaiser and F. Leisch. A toolbox for bicluster analysis in r. Technical Report 028, Ludwing-Maximilians-Universitat München, 2008. (Cited on pages 69 and 127.)

[73] M. Kanehisa and S. Goto. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000. URL http://nar.oxfordjournals.org/cgi/reprint/28/1/27. (Cited on page 20.)

[74] M. Kapushesky, P. Kemmeren, A. C. Culhane, S. Durinck, J. Ihmels, et al. Expression profiler: next generation–an online platform for analysis of microarray data. *Nucleic Acids Research*, 32(Web Server issue):W465–W470, 2004. URL http://nar.oxfordjournals.org/cgi/reprint/32/suppl_2/W465. (Cited on pages 55, 57, and 71.)

[75] M. Kaufmann, M. J. van Kreveld, and B. Speckmann. Subdivision drawings of hypergraphs. In *Graph Drawing*, pages 396–407. 2008. (Cited on page 74.)

[76] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. 2008. (Cited on pages 25, 43, 44, 45, 93, 111, and 121.)

[77] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13:703–716, 2003. (Cited on pages 53, 54, 127, and 132.)

[78] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964. URL http://www.springerlink.com/content/010q1x323915712x/. (Cited on page 69.)

[79] L. Lazzeroni and A. Owen. Plaid models for gene expression data. Technical Report, Standford University, 2002. (Cited on pages 53 and 54.)

[80] J. Liu and W. Wang. Op-cluster: Clustering by tendency in high dimensional space. In *3rd IEEE International Conference on Data Mining*, pages 187–194. 2003. (Cited on page 54.)

[81] T. Lu, Y. Pan, S. Kao, C. Li, I. Kohane, et al. Gene regulation and DNA damage in the ageing human brain. *Nature*, 429(6994):883–891, 2004. URL http://www.ncbi.nlm.nih.gov/pubmed/15190254. (Cited on pages 117 and 141.)

[82] H.-W. Ma, B. Kumar, U. Ditges, F. Gunzer, J. Buer, et al. An extended transcriptional regulatory network of Escherichia coli and analysis of its hierarchical structure and network motifs. *Nucleic Acids Research*, 32(22):6643–6649, 2004. (Cited on pages 88 and 119.)

[83] J. Mackinlay. Applying a theory of graphical presentation to the graphic design of user interfaces. In *Symposium on User Interface Software and Technology*, pages 179–189. 1988. (Cited on page 33.)

[84] S. Madeira and A. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions of Computational Biology and Bioinformatics*, 1(1):24–45, 2004. (Cited on pages 5, 51, 52, 54, 87, 90, and 91.)

[85] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, et al. Network motifs: simple building blocks of complex networks. *Science*, 298:824–827, 2002. URL http://www.sciencemag.org/cgi/reprint/298/5594/824.pdf. (Cited on page 58.)

[86] S. Mitra and Y. Hayashi. Bioinformatics with soft computing. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 36, pages 616–635. 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1678037. (Cited on page 25.)

[87] T. M. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. *Proc. Pacific Symp. Biocomputing*, 8:77–88, 2003. (Cited on pages 53, 54, 60, and 127.)

[88] C. North. Towards measuring visualization insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006. (Cited on pages 46 and 70.)

[89] L. Nowell, E. Hetzler, and T. Tanasse. Change blindness in information visualization: a case study. In *Information Visualization 2001*. 2001. (Cited on page 34.)

[90] Y. Okada, W. Fujibuchi, and P. Horton. A biclustering method for gene expression module discovery using a closed itemset enumeration algorithm. *IPSJ Digital Courier*, 3:183–192, 2007. (Cited on pages 54, 58, 59, 60, 64, 99, and 126.)

[91] H. Omote and K. Sugiyama. Force-directed drawing method for intersecting clustered graphs. In *APVis*, volume 0, pages 85–92. IEEE Computer Society, Los Alamitos, CA, USA, 2007. (Cited on pages 76, 77, and 106.)

[92] H. Omote and K. Sugiyama. Method for visualizing complicated structures based on unified simplification strategy. *IEICE Trans. Inf. & Syst.*, E90–D(10):1649–1656, 2007. (Cited on pages 75, 133, and 135.)

[93] S. Palmer and I. Rock. Rethinking perceptual organization: The role of uniform connectedness. *Psychonomic Bulletin and Review*, 1(1):29–55, 1994. (Cited on page 36.)

[94] H. Parkinson, M. Kapushesky, N. Kolesnikov, G. Rustici, M. Shojatalab, et al. ArrayExpress update–from an archive of functional genomics experiments to the atlas of gene expression. *Nucleic Acids Research*, pages 1–5, 2008. (Cited on page 17.)

[95] H. Parkinson, U. Sarkans, M. Shojatalab, N. Abeygunawardena, S. Contrino, et al. ArrayExpress–a public repository for microarray gene expression data at the EBI. *Nucleic Acid Res.*, 31(1):68–71, 2003. URL http://www.lacim.uqam.ca/~chauve/Enseignement/BIF7001/H04/Documents/Microarrays/BasesDeDonnees/MA-ArrayExpress-NAR31.pdf. (Cited on page 17.)

[96] G. A. Pavlopoulos, A.-L. Wegener, and R. Schneider. A survey of visualization tools for biological network analysis. *BioData Mining*, 1:12, 2008. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2636684. (Cited on page 117.)

[97] A. Perer and B. Shneiderman. Balancing systematic and flexible exploration of social networks. *IEEE Transactions onf Visualization and Computer Graphics*, 12(5):693–700, 2006. (Cited on page 76.)

[98] C. Plaisant, J. Fekete, and G. Grinstein. Promoting insight based evaluation of information visualization: From contests to benchmark repository. Technical Report 2004–30, Human-Computer Interaction Laboratory, University of Maryland, College Park, Maryland, 2004. (Cited on page 46.)

[99] A. Prelic, S. Bleuer, P. Zimmermann, A. Wille, P. Bühlmann, et al. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129,

2006. URL http://www.tik.ee.ethz.ch/sop/bimax/. (Cited on pages 5, 53, 54, 58, 59, 60, 65, 87, 99, 120, 125, 126, and 127.)

[100] D. A. Quigley, M. D. To, J. Pérez-Losada, F. G. Pelorosso, J.-H. Mao, et al. Genetic architecture of mouse skin inflammation and tumour susceptibility. *Nature*, 458:505–508, 2009. URL http://www.nature.com/nature/journal/v458/n7237/full/nature07683.html. (Cited on page 102.)

[101] M. Rasmussen and G. Karypis. gcluto: An interactive clustering, visualization and analysis system. Technical Report 04-021, University of Minnesota, 2004. (Cited on pages 55, 57, 64, 65, 68, 69, 70, 71, 90, 113, and 145.)

[102] T. F. Rayner, P. Rocca-Serra, P. T. Spellman, H. C. Causton, A. Farne, et al. A simple spreadsheet-based, miame-supportive format for microarray data: Mage-tab. *BMC Bioinformatics*, 7(489), 2006. URL http://www.biomedcentral.com/1471-2105/7/489. (Cited on page 21.)

[103] D. J. Reiss, N. S. Baliga, and R. Bonneau. Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. *BMC Bioinformatics*, 1:662–671, 2006. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1502140. (Cited on pages 25, 26, 59, 60, 61, and 126.)

[104] A. J. Saldanha. Java Treeview–extensible visualization of microarray data. *Bioinformatics*, 20(17):3246–3248, 2004. (Cited on pages 65, 68, 71, 81, 113, and 145.)

[105] R. Santamaría and R. Therón. Overlapping clustered graphs: Coauthorsip networks visualization. In *Lecture Notes in Computer Science. Smart Graphics 2008*, pages 190–199. Springer Verlag, 2008. (Cited on page 133.)

[106] R. Santamaría, R. Therón, and L. Quintales. Bicoverlapper: A tool for bicluster visualization. *Bioinformatics*, 24(9):1212–1213, 2008. (Cited on page 5.)

[107] R. Santamaría, R. Therón, and L. Quintales. A visual analytics approach for understanding biclustering results from microarray data. *BMC Bioinformatics*, 9(247), 2008. URL http://www.biomedcentral.com/1471-2105/9/247. (Cited on pages 5 and 43.)

[108] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):443–456, 2005. (Cited on page 47.)

[109] M. Sarkar and M. H. Brown. Graphical fisheye views. *Communications ACM*, 37(12):73–83, 1993. (Cited on pages 38 and 114.)

[110] M. C. Schatz, A. M. Phillippy, B. Shneiderman, and S. L. Salzberg. Hawkeye: an interactive visual analytics tool for genome assemblies. *Genome Biology*, 8, 2007. URL http://amos.sourceforge.net/hawkeye/. (Cited on pages 43, 82, and 83.)

[111] E. Segal, A. Battle, and D. Koller. Decomposing gene expression into cellular processes. *Proc. Pacific Symp. Biocomputing*, 8:89–100, 2003. URL citeseer.ist.psu.edu/segal03decomposing.html. (Cited on page 54.)

[112] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller. Rich probabilistic models for gene expression. *Bioinformatics*, 17:S243–S252, 2001. (Cited on page 54.)

[113] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results by interactive exploration of dendrograms, a case study with genomic microarray data. *IEEE Computer*, 35(7):80–86, 2002. URL http://www.cs.umd.edu/hcil/hce/. (Cited on pages 55 and 82.)

[114] J. Seo and B. Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. *IEEE Symposium on Information Visualization*, pages 65–72, 2004. URL http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=1382892. (Cited on pages 27, 40, 55, 57, 67, 68, 71, 82, 113, 115, and 145.)

[115] R. Shamir, A. Maron-Katz, A. Tanay, C. Linhart, I. Steinfeld, et al. Expander - an integrative program suite for microarray data analysis. *BMC Bioinformatics*, 6(232):1471–2105, 2005. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1261157. (Cited on pages 55, 57, 64, 65, 71, and 113.)

[116] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, et al. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13:24958–2504, 2003. URL http://www.genome.org/cgi/reprint/13/11/2498.pdf. (Cited on page 117.)

[117] R. Sharan and R. Shamir. Click: A clustering algorithm with applications to gene expression analysis. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 8:307–316, 2000. (Cited on page 55.)

[118] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31:64–68, 2002. (Cited on pages 88, 119, and 140.)

[119] Q. Sheng, Y. Moreau, and B. D. Moor. Biclustering microarray data by Gibbs sampling. *Bioinformatics*, 19:ii196–ii205, 2003. (Cited on page 55.)

[120] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Visual Languages*, UMCP-CSD CS-TR-3665, pages 336–343. College Park, Maryland 20742, U.S.A., 1996. URL citeseer.ist.psu.edu/shneiderman96eyes.html. (Cited on pages 31 and 110.)

[121] B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In *Information Visualization*, pages 73–78. 2001. URL http://ieeexplore.ieee.org/iel5/7626/20791/00963283.pdf?tp=&isnumber=&arnumber=963283. (Cited on page 76.)

[122] H. Siirtola. Combining parallel coordinates with the reordable matrix. *Information Visualization*, 4(1):32–48, 2003. URL http://infoviz.cs.uta.fi/publications.php. (Cited on page 40.)

[123] S. S. Stevens. On the theory of scales of measurement. *Science*, pages 677–680, 1946. (Cited on page 33.)

[124] K. Sugiyama and K. Misue. Visualization of structural information: automatic drawing of compound digraphs. *Systems, Man and Cybernetics, IEEE Transactions on*, 21(4):876–892, Jul/Aug 1991. (Cited on page 76.)

[125] P. Tamayo, D. Slonim, J. Mestrov, Q. Zhu, S. Kitareewan, et al. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, 96:2907–2912, 1999. (Cited on page 26.)

[126] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneus genomewide data. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2981–2986, 2004. (Cited on pages 5, 53, 54, 55, and 60.)

[127] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18:S136–S144, 2002. (Cited on pages 51, 53, 54, and 88.)

[128] C. Tang, L. Zhang, and M. Ramanathan. Interrelated two-way clustering: An unsupervised approach for gene expression data analysis. In *In Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, pages 41–48. 2001. (Cited on page 54.)

[129] R. Therón, R. Santamaría, J. García, D. Gómez, and V. Paz-Madrid. Overlapper: movie analyzer. In *Information Visualization Confererence Compendium*, pages 140–141. 2007. URL http://conferences.computer.org/infovis/files/compendium2007.pdf. (Cited on page 133.)

[130] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Press, 2005. (Cited on pages 4, 43, 44, 45, and 47.)

[131] R. Tibshirani, T. Hastie, M. Eisen, D. Ross, D. Botstein, et al. Clustering methods for the analysis of DNA microarray data. Technical Report, Dept. of Health Research and Policy, Dept. of Genetics, Dept. of Biochemistry, Stanford Univ., 1999. URL citeseer.ist.psu.edu/tibshirani99clustering.html. (Cited on page 54.)

[132] E. Tufte. *The Visual Display of Quantitavie Information*. Graphics Press, 1983. (Cited on page 31.)

[133] E. Tufte. *Envisioning Information*. Graphics Press LLC, 1990. (Cited on page 30.)

[134] H. Turner, T. Bailey, and W. Krzanowski. Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational Statistics and Data Analysis*, 48:235–254, 2003. (Cited on pages 58, 99, and 127.)

[135] H. L. Turner, T. C. Bailey, W. J. Krzanowski, and C. A. Hemingway. Biclustering models for structured microarray data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):316–329, 2005. (Cited on pages 53, 54, 59, 125, 127, and 132.)

[136] L. J. van 't Veer, H. Dai, M. J. van de Vijver Y. D. He, A. A. Hart, M. Mao, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Letters to Nature*, 415:530–536, 2002. URL http://www.nature.com/nature/journal/v415/n6871/abs/415530a.html. (Cited on pages 26 and 64.)

[137] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, 2002. (Cited on page 24.)

[138] J. C. Venter and et al. The sequence of the human genome. *Science*, 291:1304–1351, 2001. URL http://www.sciencemag.org/cgi/reprint/291/5507/1304.pdf. (Cited on page 3.)

[139] A. Verroust and M.-L. Viaud. Ensuring the drawability of extended Euler diagrams for up to 8 sets. In *Diagrammatic Representation and Inference: Third International Conference*. 2004. URL http://www-rocq.inria.fr/imedia/Articles/VerroustViaud.pdf. (Cited on page 75.)

[140] F. B. Viegas and M. Wattenberg. Tag clouds and the case for vernacular visualization. *Interactions*, 15(4):49–52, 2008. URL http://portal.acm.org/citation.cfm?id=1374501. (Cited on page 116.)

[141] C. Ware. *Information Visualization: Perception for Design*. Diane Cerra, 2nd edition, 2004. (Cited on pages 4, 29, 31, 33, 34, 35, 38, 64, 66, 74, 107, and 109.)

[142] J. M. Wolfe, N. Klempen, and K. Dahlen. Postattentive vision. *Journal of Experimental Psychology*, 26(2):693–716, 2000. (Cited on page 34.)

[143] V. Wood, R. Gwilliam, M. A. Rajandream, M. Lyne, R. Lyne, et al. The genome sequence of Schizosaccharomyces pombe. *Nature*, 415:871–880, 2002. URL http://www.nature.com/nature/journal/v415/n6874/abs/nature724.html. (Cited on page 3.)

[144] C. J. Wu and S. Kasif. Gems: a web server for biclustering analysis of expression data. *Nucleic Acids Research*, 33:596–599, 2005. (Cited on pages 55 and 57.)

[145] J. Yang, W. Wang, H. Wang, and P. Yu. d-clusters: Capturing subspace correlation in a large data set. In *Proc. 18th IEEE Int'l Conf. Data Engineering*, pages 517–528. 2002. (Cited on pages 23 and 54.)

[146] J. Yang, W. Wang, H. Wang, and P. Yu. Enhanced biclustering on expression data. In *Third IEEE Conf. Bioinformatics and Bioengineering*, pages 321–327. 2003. (Cited on page 54.)

[147] W. Zhang, A. Collins, N. Maniatis, W. Tapper, and N. E. Morton. Properties of linkage disequilibrium (LD) maps. *Proceedings of the National Academy of Sciences of the United States of America*, 99(26):17004–17007, 2002. URL http://www.pnas.org/content/99/26/17004.abstract. (Cited on page 79.)

COLOPHON

This thesis was typeset with LaTeX 2ε using Hermann Zapf's *Palatino* and *Euler* type faces (Type 1 PostScript fonts *URW Palladio L* and *FPL* were used). The listings are typeset in *Bera Mono*, originally developed by Bitstream, Inc. as "Bitstream Vera". (Type 1 PostScript fonts were made available by Malte Rosenau and Ulrich Dirr.)

The typographic style was inspired by Bringhurst's genius as presented in *The Elements of Typographic Style* [22]. It is available for LaTeX via CTAN as "`classicthesis`".