Development and application of methods for the analysis of microarray gene expression data

Bjarte Dysvik

PhD thesis

Department of Informatics
University of Bergen
2006

ACKNOWLEDGMENTS

This thesis is based upon studies conducted during November 2002 to June 2006 at the Department of Informatics, University of Bergen, Norway.

First of all I would like to express my sincere gratitude to my supervisor, professor Inge Jonassen. Without his advice, patience and knowledge, this thesis would never have become a reality. Further I would like to thank my immediate microarray group at the university for their many hours of fruitful discussions and positive attitude. Specifically I would like to thank Trond Hellem Bø, Kjell Petersen, Anne-Kristin Stavrum and Laila Stordrange who all deserve much credit for this work.

I would also like to thank all of my collaborators from the many different projects in which I have participated. This includes Sala O. Ibrahim, Mai Lill Suhr and Endre N. Vasstrand from the Department of Biomedicine and Dental Faculty-Periodontology at the University of Bergen; Petter Frost, Christiane Moros and Frank Nilsen from the genomics group, Institute of Marine Research, Bergen, Norway and Frédéric Pendino from the Department of Molecular Biology, University of Bergen.

I would like to thank the Research Council of Norway that has funded this work through the Salmon Genome Project and the functional genomics program FUGE and its technology platform for microarrays.

Additionally, I would like to thank several people helping with J-Express and MolMine. These are Kristin Sandereid, Øivind Enger and Erlend Skagseth from Sarsia Innovation and Vidar M. Steen from the Centre for Medical Genetics and Molecular Medicine, Haukeland University Hospital.

Finally, but not least, I wish to thank my family who have always supported me, my good friends in Stavanger and most of all Wenche for enjoying life together with me.

Bjarte Dysvik, Bergen June 2006.

Table of contents

1 Introduction

The use of high-throughput technologies in molecular biology has opened the way to a post-genomic era. Scientists are no longer limited to study just a handful of genes or proteins at the time, but can now screen full genomes and study complete biological systems more efficiently than ever before. The introduction of microarrays has revolutionized the way gene expression studies are performed and is already leading to important medical discoveries. The technology is however still considered to be in its infancy, with many problems still to be solved.

High throughput generally means a lot of data, which needs to be organized and analyzed in an effective manner. Most available technologies for generating large amounts of biological data such as microarrays, 2d-gels and mass spectrometry focus on quantity at the expense of accuracy. This, together with the fact that it is hard to effectively store and make sense of millions of measurements for a single experiment, makes the use of computers unavoidable. Bioinformatics is an emerging field where informatics and biology join forces by applying informatics expertise to biological problems.

This thesis will focus on two topics. First the analysis of new proprietary data produced by collaborators, and second, the development of new methods for high throughput data analysis and preparation. The goal is to increase knowledge and understanding of microarray technology and use this knowledge to develop novel methods for improving the quality of microarray results.

Biology and biological systems

This section is mainly based on two sources: [1, 2].

1.1 The central dogma of molecular biology

The central dogma in molecular biology describes the process in which information stored in a DNA molecule is *transcribed* to form an mRNA molecule and further *translated* to a protein. A protein in its simplest form is a chain (sequence) of amino acids (some 20 different ones), each of which has very specific properties such as charge and size. Each amino acid corresponds to a triplet (three succeeding bases) in the DNA sequence. By composing the proteins of specific amino acid sequences, they get specific properties which determine the proteins' function. Some proteins are used as structural components and building blocks, while others have more active roles such as enzymes and regulatory proteins (see Figure 1). For a cell to create a certain protein, the DNA sequence corresponding to the protein must be correct and both the transcription and translation machinery must be accurate. A single error in the transcribed sequence may result in total function loss for the new protein.

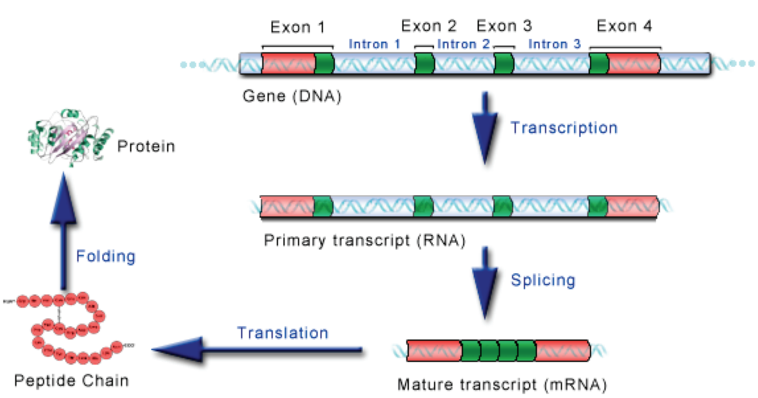


Figure 1: From genes to proteins: between coding *exons* are non-coding *introns* which are spliced out before translation. Such introns are found in most eukaryotes (cells with a nucleus).After translation, the chain of amino acids is folded to form a functioning protein (here represented by a cartoon model).

Genetic mutations are changes in the DNA caused by events such as copying errors or radiation. These can be harmless if repaired by the cells genetic repair apparatus or if they occur in non-functional areas, but problematic and even fatal if occurring within the coding or regulatory area of a gene (for instance sickle cell anemia which is caused by a single mutation at codon 6 of the β-globin gene).

For a cell to function properly, it needs to produce the correct proteins in the correct amount as the surroundings and internal environment changes. This process is called gene regulation and takes place at several different levels, including synthesis of RNA transcripts, posttranscriptional processing of mRNA, mRNA and protein degradation, translation, posttranslational protein modification and protein transport. Gene regulation is influenced by various mechanisms, such as outside signals, stress response and the cell cycle. For instance, the expression of certain enzymes may increase or decrease as the organism's food sources change or are depleted.

Gene regulation defects can be fatal for a cell and is the cause of many diseases. Some proteins operate as protein complexes or in functional assembly lines where the product of one protein is the source of another (e.g metabolic pathways [3]) and therefore depend on joint regulation. Regulation failure in a single component may result in a completely useless protein complex or a metabolic pathway with a devastating bottleneck [4].

By using new technology, it is now possible to monitor the expression change for thousands of genes simultaneously, and thereby effectively spot regulation disorders. This technology will be thoroughly discussed in the coming chapters.

1.2 Functional genomics and systems biology

Functional genomics and systems biology [5] (A portal for systems biology: http://www.systems-biology.org) are some of the hot topics in molecular biology today. New technologies make it possible to look beyond the expression of single genes, and even single regulatory networks [6] and try to understand how the complete system of a cell behaves. This is accomplished by building models based on knowledge about single genes [7], gene interactions and regulatory processes. These models are constantly refined through scientific experiments to increase prediction rates. The ultimate goal is to be able to understand the entire biological system and use this to predict how an organism reacts to a certain stimuli without going to the laboratory at all, but simply input the stimuli as a parameter to a computer model. A predictive computer model is also a proof of a true understanding of the system. This is the field of systems biology; an interdisciplinary field studied by computational biologists, statisticians, mathematicians, engineers, physicists and computer scientists. One important factor in this field is the use of computers to analyze, organize, store and query large quantities of data generated by high throughput methods. In addition, computers are used to build models and simulate systems to verify or reject additional hypotheses.

A biological system can be simulated with a mathematic model. To test the model, effects of perturbations such as gene knockouts can be predicted mathematically and compared to real life perturbations. When testing new hypotheses, a sufficiently accurate model can then be used as a filter for selecting the most promising experiments.

Systems biology can be divided into four problem areas. These are: (1) understanding the structure of the systems, such as genes, signal transduction and metabolic pathways, (2) learning the dynamics of such systems, (3) developing methods to control the systems and (4) developing methods for designing and modifying new systems for desired properties.

The system structures can for instance be regulatory relationships of genes, protein interactions or physical structures of cells. The system dynamics describes how a system behaves over time and responds to external stimuli. Some refer to these two first areas as “parts lists” and “connections” and they are the constituent parts of the system. Controlling or building new systems is often the ultimate goal. For instance, when knowledge about the processes leading from a normal cell to a cancer cell is understood and an accurate model is established, the next step is to apply methods to reverse this process and have the cell return to the normal state or enter apoptosis (programmed cell death).

The actual technologies needed for systems biology to be a realistic science will be discussed in the next chapter, but a key factor is comprehensiveness. For instance, the complete genome of baker’s yeast (*Saccharomyces cerevisiae*) was sequenced in 1996, and shortly after a microarray for analyzing mRNA expression for nearly all open reading frames (potential proteins) became available (Patrick O. Brown’s laboratory at Stanford University. http://cmgm.stanford.edu/pbrown). Using this microarray, scientists could now measure gene regulation, build and test models based on gene expression measurements for all potential genes

Metabolic pathways

Systems biology can for example be used to model metabolic pathways [8]. Figure 2 shows how the KEGG database (http://www.genome.jp/kegg) represents the citrate cycle. This pathway shows how enzymes work in an assembly-line like fashion to extract energy by converting metabolites from high energy containing molecules to lower energy containing molecules.

Modeling pathways like the citrate cycle can be done in a bottom-up approach where the individual components are identified and studied before the “global” structure of the system is formed and tested. When a new organism is studied, existing models can be applied in a top-down approach where individual components are identified by for example sequence similarity methods.

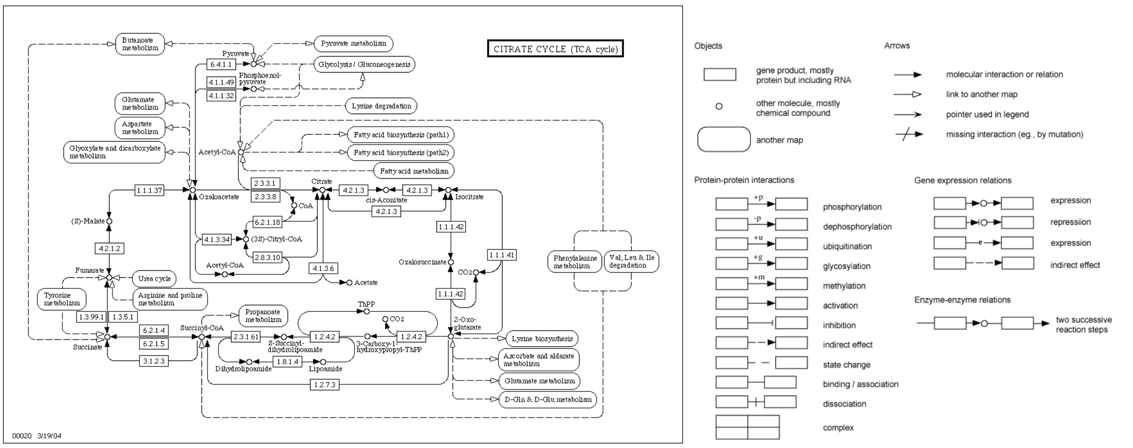


Figure 2: The citrate cycle (left) and the description of structural components and interactions (right). Charts generated by the KEGG database (Kyoto Encyclopedia of Genes and Genomes, release 36.0 http://www.genome.jp/kegg/pathway/map/map00020.html).

The MetaCyc [9] database is another database with systems biology information. It contains over 700 metabolic pathways curated from scientific experimental literature and combines biological knowledge like pathway information (including reactions and compounds) with genes and protein products. It is also possible to query the MetaCyc database using sequence information to look for known pathways and genes in new less studied organisms in a bottom-up like fashion.

Gene nomenclature and Gene Ontology

For a description of a biological system like the citrate cycle to be of any value to other scientists than its discoverers, it is important that the components making up the system are uniquely defined. Many efforts have been made towards creating one single standardized nomenclature for genes, but scientists are still using several of them simultaneously to be sure the genetic component is uniquely recognized. In its simplest form, a gene could be identified by its sequence. A single gene can however go trough post-translational modifications, and thus end up as different functional molecules, which makes gene identification a challenge. Polymorphisms, splice variants, mutations, functions and even orthologs in other sequenced organisms are valuable information that must be accessible, preferably in centralized databases. Some of the most accessed databases with such data publicly available are GenBank (http://www.ncbi.nlm.nih.gov/Genbank) and UniGene (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene).

Gene Ontology [10-12] is a project that aims at defining an accurate, dynamic and controlled vocabulary which can be applied to gene and protein roles. It is dynamic in the sense that it is not limited to certain organisms. This is particularly important when transferring biological roles between organisms, such as searching for homologous genes and pathways in less studied species. The Gene Ontology is divided into three areas: *biological process*, *molecular function* and *cellular component*. Directed acyclic graphs for these three groups can be downloaded and mapped to many genomes. The graphs are organized in a form where nodes go from "more general" to "more specific". For instance, *Biological_process* is a top node with a child called *behavior* which again has child nodes *adult behavior*, *auditory behavior* and *behavioral fear response*.

- all : all (<367946)
  - GO:0008150 : biological_process (<128040)
    - GO:0000004 : biological process unknown (<33969)
    - GO:0009987 : cellular process (<78831)
      - GO:0007155 : cell adhesion (<1461)
      - GO:0007154 : cell communication (<12283)
      - GO:0030154 : cell differentiation (<3552)
      - GO:0008037 : cell recognition (<68)
      - GO:0050875 : cellular physiological process (<72062)
      - GO:0050794 : regulation of cellular process (<12772)
    - GO:0007275 : development (<13755)
    - GO:0040007 : growth (<3303)
    - GO:0051704 : interaction between organisms (<1443)
    - GO:0007582 : physiological process (<81248)
    - GO:0043473 : pigmentation (<98)
    - GO:0050789 : regulation of biological process (<15932)
    - GO:0000003 : reproduction (<4324)
    - GO:0050896 : response to stimulus (<15834)
    - GO:0016032 : viral life cycle (<306)
  - GO:0005575 : cellular_component (<117135)
  - GO:0003674 : molecular_function (<122771)

Figure 3: Hierarchical representation of a small part of the GO structure (*biological process* and its child-node *cellular process* have been opened). The chart is generated by the AmiGO browser (http://www.godatabase.org/cgi-bin/amigo/go.cgi). A line in this structure contains the following parts: The first icon (plus or minus) shows if a node has child-nodes and if it is open (minus). The green I and pink P shows the type of relation to the parent node (is_a and part_of relations respectively). The next is the ID and name of the GO term. The number in parentheses on the end is the number of gene products associated with the GO term using a predefined database.

1.3 High throughput technologies for molecular biology

It has become fashionable to invent new terms to describe the global set of biological molecules or phenomena that is studies. Established terms include genome (the complete set of genes, or DNA of an organism), transcriptome (the complete set of transcripts), proteome (the complete set of proteins) and metabolome (complete set of metabolites). Studies and measurements of the various -omes are referred to as corresponding -omics (e.g. genomics, proteomics, transcriptomics, metabolomics) [13] and often involve use of high throughput technologies.

The development of these technologies has happened in parallel with computers becoming increasingly powerful and affordable. The increased availability of computing power has been a prerequisite for the technology development and deployment. New algorithms, modeling techniques and specialized software including databases and data processing functionality are constantly contributing to the growth of bioinformatics. In the following section we describe some of the high throughput technologies and their corresponding (predecessor) low throughput counterparts.

High Throughput	Low Throughput Equivalents	Compound measured
Gel electrophoresis	Western blot (http://en.wikipedia.org/wiki/Western_blot), Chromatography	Proteins / DNA /mRNA
Mass spectrometry	Western blot, Chromatography	Proteins, metabolites
DNA microarrays	Real time RT-PCR, SAGE (Serial Analysis of Gene Expression), Northern Blot (mRNA) [14], Southern blots (DNA) [15]	RNA / DNA
Protein/Antibody arrays	Western blot, chromatography	Proteins

Table 1: Some of the most relevant high throughput technologies used in molecular biology and their low throughput counterparts

Technologies based on molecule separation

Gel electrophoresis is a method for separation of macromolecules, either nucleic acids or proteins on the basis of size, electric charge or other physical properties. 2D SDS- PAGE (2-Dimensional Sodium Dodecyl Sulfate – PolyAcrylamide Gel Electrophoresis) is probably the most widely used method for separating and identifying proteins and their abundance. The method separates proteins based on size in one direction and isoelectric point [2] in the other. After separation, proteins can be visualized by conventional staining techniques. For further identification of the separated molecules, the spots can be cut out of the gel and further processed by for instance mass spectrometry (ms) to identify the protein. Proteins can also be identified by mapping their location on the gel to a size-isoelectric point library.

Mass spectrometry is a method that can be used for separating molecules (e.g ionized proteins or peptides) based on their mass to charge ratios. Proteins in a sample are normally converted to peptides (short protein sequeces) using proteolytic enzymes. The peptides are then separated and subjected to mass spectrometry to identify and/or quantify them. It is also possible to apply the separation process first and then convert the separated proteins into peptides. The sequence of the peptides can be identified by mapping them to known mass/charge libraries or by using a method called tandem mass spectrometry (also referred to as ms-ms). Other ways of performing mass spectrometry include MALDI-TOF MS, LC-MS/MS and Ion trap-MS (the prefix is referring to different ways of generating charged peptides).

Technologies based on molecule attachment

DNA Microarrays will be discussed thoroughly in the next chapters, but is briefly described here for comparison purposes. They consist of cDNA molecules (or oligonucleotides) with known sequences, referred to as probes, attached to a medium. These probes will attach to complementary labeled target sequences (mRNA or DNA) in the studied sample. When bound to a probe, a target molecule will emit a signal when the array is scanned. The strength of this signal will reflect the abundance of the bound type of molecule (e.g. abundance of a certain gene).

Protein microarrays [16-18] are based on the same principles as DNA microarrays in that a "bait" molecule is printed on a solid medium and labeled target molecules are allowed to bind. Abundance is then confirmed and measured by label emittance. Baits are molecules that bind to specific proteins, for instance antibodies or parts of protein complexes.

Many of the high throughput methods used in molecular biology have their strength in the number of measurements performed simultaneously, but often lack the accuracy associated with their low throughput conventional counterparts. For many experiments, the objective is to identify small sets of genes or proteins responsible for phenotypical differences. To increase confidence, interesting results from high throughput experiments should be verified by more accurate low throughput methods, for example quantitative real-time PCR [19] and Northern blot hybridization.

2 Microarrays

Microarray technology is very diverse with many variants and applications [20, 21]. Two of the first microarrays are the first Genechip® from Affymetrix [22] (www.affymetrix .com; Figure 5 and 6) and the first *Arabidopsis* microarray [23] from Patrick O. Brown's laboratory at Stanford University. These represent early versions of two different ways of producing a microarray (on-array synthesis using photolithography and printing of cDNAs/PCR products).

Microarrays commonly contain a large number of elements of the same type in a relatively small area. Each element can be a probe for a particular molecule species, e.g. the probes can be complementary (in the Watson-Crick sense) to labeled poly-nucleotides (typically cRNA or cDNA) or antibodies for a particular protein species. Another type of microarray contains miniaturized 'laboratories' so that a large number of identical experiments can be performed in parallel within a small area (lab on a chip, see e.g. http://www.rsc.org/Publishing/Journals/lc)

Microarrays are produced by printing or synthesizing the probe molecules in separated areas on a solid surface such as glass or nylon. Figure 4 shows how to use a two-channel printed cDNA microarray to find genes differentially expressed between a test sample and a reference sample. In literature, there have been some disagreements between the terms *probes* and *targets*. In this text we will follow the suggested definitions from "The Chipping Forecast" (a supplementary to Nature genetics, January 1999, Volume 21) where probes are the molecules immobilized on the microarray substrate and targets are the molecules whose abundance is measured. In addition, we refer to the printed probes as genes, while in practice the actual molecules measured can be other kinds of molecules such as DNA segments. Finally, we refer to areas on the microarray where a certain molecule is measured as a "spot", while in general the area may not resemble spots at all (such as for Affymetrix arrays).

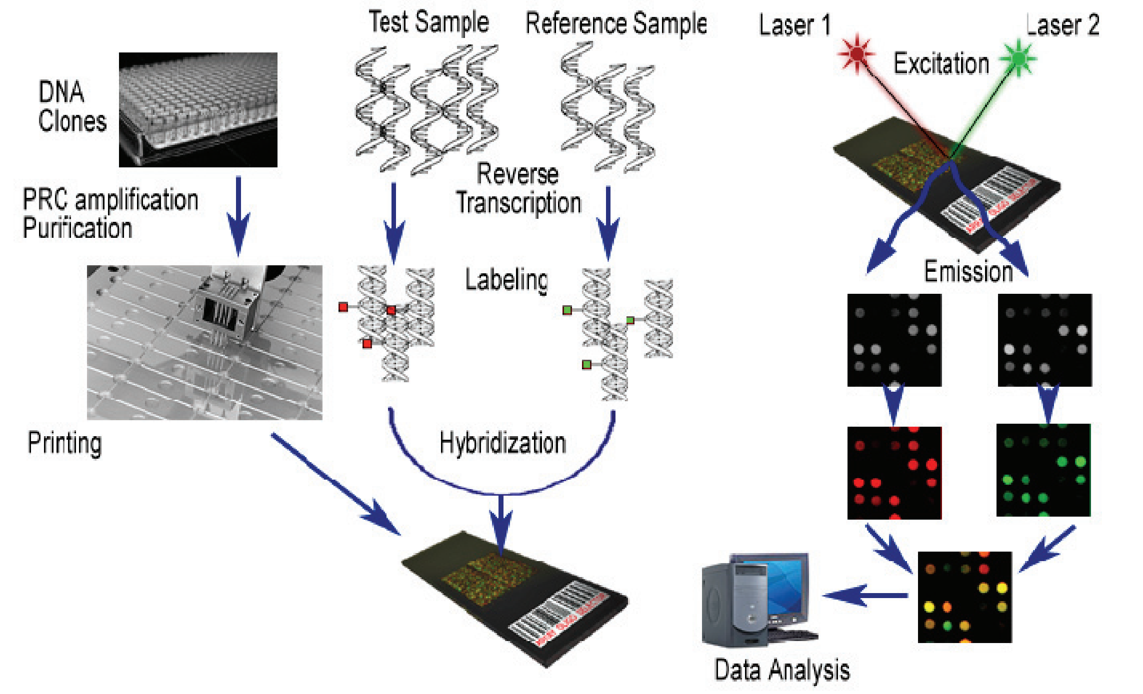


Figure 4: Outline of a typical two-channel microarray study. The left side outlines the printing process, the middle represents labeling and hybridization and the right side represents scanning and image/data analysis.

This thesis will focus on the analysis of data from mRNA-measuring microarrays, but many of the challenges and problem areas are common for most microarray technologies. Different types of mRNA arrays exist, but the major versions are those synthesized base by base directly on the array (in situ synthesized arrays such as the Genechip ® from Affymetrix, see Figure 5), those synthesized off-chip and printed in spots (pre-synthesized oliogo arrays) and those created by printing spots of cDNA (cDNA-arrays). The differences in price and precision have long made Affymetrix arrays a preferred technology for the industry and cDNA arrays for academic institutions [24, 25], but new companies have started to deliver accurate arrays for academic labs (e.g Applied Biosystems and Agilent). cDNA arrays have to some extent been replaced by oligo arrays because of their seeming lack of precision, but are still frequently used, much because they allow construction of relatively cheap custom arrays for less studied organisms.

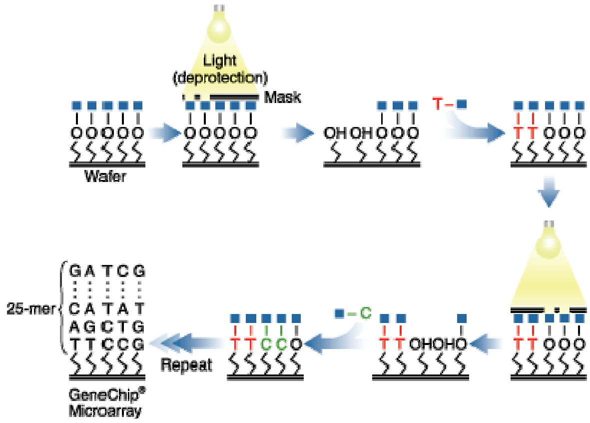


Figure 5: The photolithography synthesis process for Affymetrix Genechips ® (Figure from Affymetrix web site).

Figure 4 shows a typical two-channel microarray experiment outline. mRNA is extracted from a test sample, reverse transcribed and labeled with a dye. The same procedure with a different dye is performed for another sample (often a common reference for multiple array studies) and both samples are then exposed to a microarray where the labeled target molecules will hybridize to the printed probes. By scanning the array at different wavelengths (corresponding to the label emittance), a comparative signal for the abundance of certain transcripts can be found.

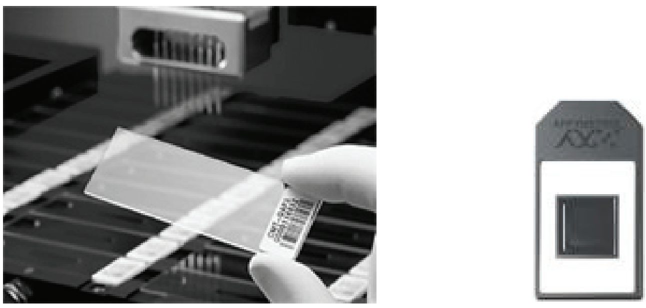


Figure 6: Two common types of microarrays. A custom cDNA array printed on glass (left) and an Affymetrix Genechip ® (right).

2.1 Practical use of microarrays

In an early study by Alizadeh et. al. [26] microarrays were used to analyze the expression profiles of several thousand genes in large B-lymphoma, a malignant cancer in the lymphatic system. In this study, the authors discovered two different groups of patients based on the gene expression patterns. By looking at the mortality rates, it became clear that these groups had significantly different survival rates. The same study also revealed gene groups with different expression patterns between the patient groups. The outcome of this study shows the potential of microarray technology in regards to diagnosis, prognosis as well as opportunities towards therapeutics and biomedical research [27]. Similar studies have revealed analogous results for other genetic related diseases, such human breast [28] and skin tumors [29], leukemia [30], colon cancer [31], prostate cancer [32], small round blue cell tumors (SRBCTs) [33] and brain tumors [34].

Besides cancer studies, microarrays has been used in a variety of problem areas within functional genomics. Derisi et. al [21] used a microarray to monitor expression change in yeast as it changed its metabolism from fermentation to respiration.

Customized arrays have been used to discover genetic alterations such as sequence variation [35] and screening of genomic imbalances, e.g. genetic amplifications and deletions (Comparative Genomic Hybridization arrays, CGH) [36, 37]. Gene expression in relation to growth and development has been studied for organisms such as *drosophila* [38] and malaria [39] to find genes turned on and off in metamorphosis and other steps of the life cycle. Understanding the developmental regulation of these genes can lead to effective drug therapies by blocking regulation of important genes to prevent maturity and propagation of for instance malaria and malaria carrying mosquitoes.

The examples mentioned here are only a few of a large and growing number of studies using microarray technology. A search in PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed) with the term 'microarray' returns 12932 hits at the time this was written (7 june 2006). Approximately 50% of these were added since 1.1.2005. In addition, the query 'microarray and cancer' return 4562 hits underlining the importance of this technology in cancer research.

2.2 Designing microarray gene expression experiments

The first and probably most important step in setting up a microarray experiment is to select an experimental design [40-47] that will answer the correct questions at an affordable cost. Typical questions that can be answered with microarrays are:

- What is the difference in gene expression between normal tissue and disease tissue?
- What is the gene expression difference between disease tissues in different stages?
- Which genes are changing in expression as an organism grows?
- Which genes are changing in expression when a particular gene is knocked out?
- Which genes are changing in expression as an organism changes form (metamorphosis)?
- Which genes are changing in expression when a drug is injected, and how much do they change?
- Which parts of the genome are transcribed?
- Are there any genetic deletions or duplications in chromosomes from a tumor sample compared to healthy chromosomes?

The design of a simple microarray experiment can be divided into three different layers [47]. In the top layer, biological objects (such as mice, patients, cell lines etc.) are assigned to variant groups (treated, not treated, disease state etc). In the middle layer, mRNA is extracted from each of the biological objects and labeled. The labeled samples are then hybridized to their treatment counterparts in a way that maximizes information about the biological question at hand. The bottom layer involves array design and the physical layout of spots. The many sources of variation in a typical microarray experiment can be distributed among these three layers. Biological variation occurs in the top layer and is often the main focus of the experiment. By carefully choosing the samples to include in the experiment, it can be possible to correlate treatment directly to gene expression, and rule out gene expression caused by other genetic or environmental factors. Technical variation appears in the mid-layer and is introduced in every step from obtaining the samples to fixing the molecules to a microarray. This includes extraction, labeling and hybridization. Measurement error is introduced in the bottom layer and is associated with reading the signals emitted by the labeled molecules.

One of the fundamental questions when defining a threshold of accuracy needed to obtain meaningful microarray results is when, where and how to use replicates. More replicates should normally lead to better results, but the cost of microarrays, chemicals and work-hours requires scientist to balance cost, confidence and efficiency against the desire to explore more experimental conditions. The variance from the two higher layers is usually the target for most microarray experiments and replications should be incorporated in such a way that this variance can be addressed. In a publication in Nature [47], Gary A. Churchill claims that correlation between replicate spots on a single microarray will normally exceed 95%, that correlation between spots on two microarrays with the same hybridized material is likely to fall between 60 and 80% and that the correlation between spots on two microarray hybridized from individual inbred mice may be as low as 30%. Although this paper was published in 2002, and recent studies suggest that this number may be somewhat higher (and probably very dependent on the technology used), it demonstrates the problem with reproducing microarray studies. It also shows that it is possible to reduce variation in the experiment by limiting the samples to come from fewer biological replicates. The power of scientific hypotheses and conclusions made from such experiments will however be lower than those based on more biological replicates simply because they are less likely to be reproducible.

Replicate	Addresses
Identical probes on same chip	Measurement error
Same labeled target hybridized on two or more arrays	Technical variance. Hybridization and quantification effects
Same sample re-labeled	Technical variance. Labeling effects
New extraction of targets from the same source sample	Technical variance. Extraction effects
Different source from the same treatment group (treated, non-treated, disease etc.)	Biological individual variation

Table 2: Examples of replicates and what they address.

Hybridization design and chip design are two important parts of a microarray experiment. Hybridization design deals with questions such as which samples to hybridize, use of biological replicates and sample size. Chip design involves everything from chip material to the actual immobilization of probes to the chip surface and deals with important challenges such as probe layout, probe density and probe sequence. Probe replication is simply repeated spotting of the same probes in different locations on the array which increases precision of the measurements. Printing multiple spots of the same probe is often called within-array replication and is a relatively cheap form for replication as long as there is physical space on the array. To measure effects such as local background fluctuations in the foreground signal, these should be placed randomly across the array instead of side by side. If other sources of variation are present on the array (such as pin-groups or sub-arrays), assignment to these should be randomized to prevent confounding effects.

Microarray chip design involves:

- Chip substrate selection
- Probe attachment mechanisms
- Selection of spot size, between spot distance and spot layout (e.g. quadratic or diamond layout)
- Physical array size
- Number, size and arrangement of sub-arrays
- Type, arrangement and number of controls on the array

Hybridization design involves:

- Samples to hybridize
- Hybridization schemes (pairwise hybridization, common reference hybridization etc. (see Figure 8)).
- Use of replicate arrays
- Use of dyes/emittance
- Number of samples and hybridizations

- Order of hybridizations
- Batches of hybridizations
- Personnel (experimentalist, technicians etc.)

Technical variance and measurement error can reduce or even completely obscure the biological variance. The sources of technical variation and measurement errors, often referred to as nuisance effects or nuisance factors, are many and not completely understood. Some are however identified (such as dye and array effects) and should be dealt with prior to downstream expression analysis. Other sources, such as lab habits are more difficult to control. Many factors are also influencing each other. The potential number of variation sources for as few as 4 factors could result in as many as $2^4=16$ possible experimental effects [46]. For an experiment with the factors Arrays A, Dyes D, Varieties V (factors of interest), and Genes G, there are four direct effects A, D, V and G, six two factor effects, AD, AV, AG, DV, DG and VG, four three factor effects, ADV, ADG, AVG and DVG, and one four factor effect ADVG. The experiment should be designed in such a way that none of the known nuisance effects are confounding with the experiment objectives. For instance, if the design is a paired design using a two-channel array with normal samples in one channel and disease samples in another, the dye effect confounds disease-state and it will be difficult or impossible to determine how much of the signal is caused by dye effect and how much is caused by biology. Instead, the samples should be balanced between the two dyes so that an equal number from each experimental state or treatment group is labeled with each dye. Dye-swap is a popular approach to reduce technical variance caused by dye effects in two-channel microarrays, and prevent this from confounding with the experimental objectives. Dye swaps can be applied to technical replicates so that each sample is hybridized two times, one with each dye. For direct hybridizations such as treatment vs. control, this means two arrays to compare two samples.

Random sampling from experimental groups is important for the validity of the statistical test used in a microarray experiment [47].True random sampling is hard to achieve, but a good representative selection is often obtainable. Many confounding effects can be removed by randomization. When arrays from different batches are used, one could randomly choose an array from one of the batches for each sample to prevent batch to confound with the treatment group. Similarly, for each sample, a random dye assignment can prevent dye biases to confound with treatment group.

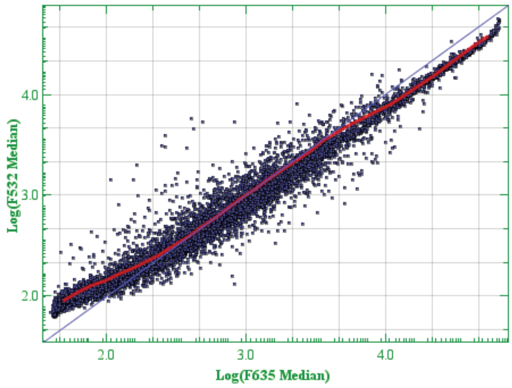


Figure 7: Log-log plot of the two channels in one two-channel microarray. Non-linear effects make normalization between channels more than a matter of simple linear scaling. As intensity increases, the mean ratio between the two channels also changes (F532 and F635 are the wavelength of the two dyes used for this array). The red line is a trend line showing the mean signal across diagonal windows in the log-log plot.

Popular hybridization designs for two-channel arrays are the reference sample design, the pooled reference sample design, the loop design and the pairwise design [47, 48] (see Figure 8). In the reference sample design, one of the samples is used as reference channel for all other samples (often a “normal” or “time 0” sample). Although very simple, there are some major concerns for this design. For instance, it is a problem if the reference lacks a good signal (good being well above background signal) for probes expressed in other samples, as low signals are more often influenced by noise. In addition, two-channel arrays are often combined to a single ratio or log ratio. If the denominator of the ratio is low and significantly influenced by noise, the result is much less trustworthy than with a strong denominator. The pooled reference reduces this problem by pooling many samples to create a strong reference base signal for most probes. For both common reference designs, the transitive nature of the signal comparison between two samples implies more experimental noise than a direct comparison. In addition, using one or more of the samples to create a reference sample causes disadvantageous dependencies between signals in the two channels. For the pooled reference, the relative signal in all samples will depend on the abundance of signals in all the other samples which can cause problems for many common statistical assumptions. With the loop and pairwise designs it is possible to circumvent this problem by direct hybridization the samples one wish to compare. A common problem with these designs however, is that whenever one array is bad or damaged, other arrays in the model may also be affected. In addition, while reference design experiments can be quite easily analyzed in the form of an expression matrix, loop and pairwise designs often require quite advanced analysis methods such as ANOVA and Bayesian models. An exception is the direct pairwise design where each sample is hybridized to a matched counterpart, such as disease tissue against normal tissue from the same patient. A disadvantage with this design, is that it does not allow comparison between patients (ratios may be compared, but not individual sample channels).

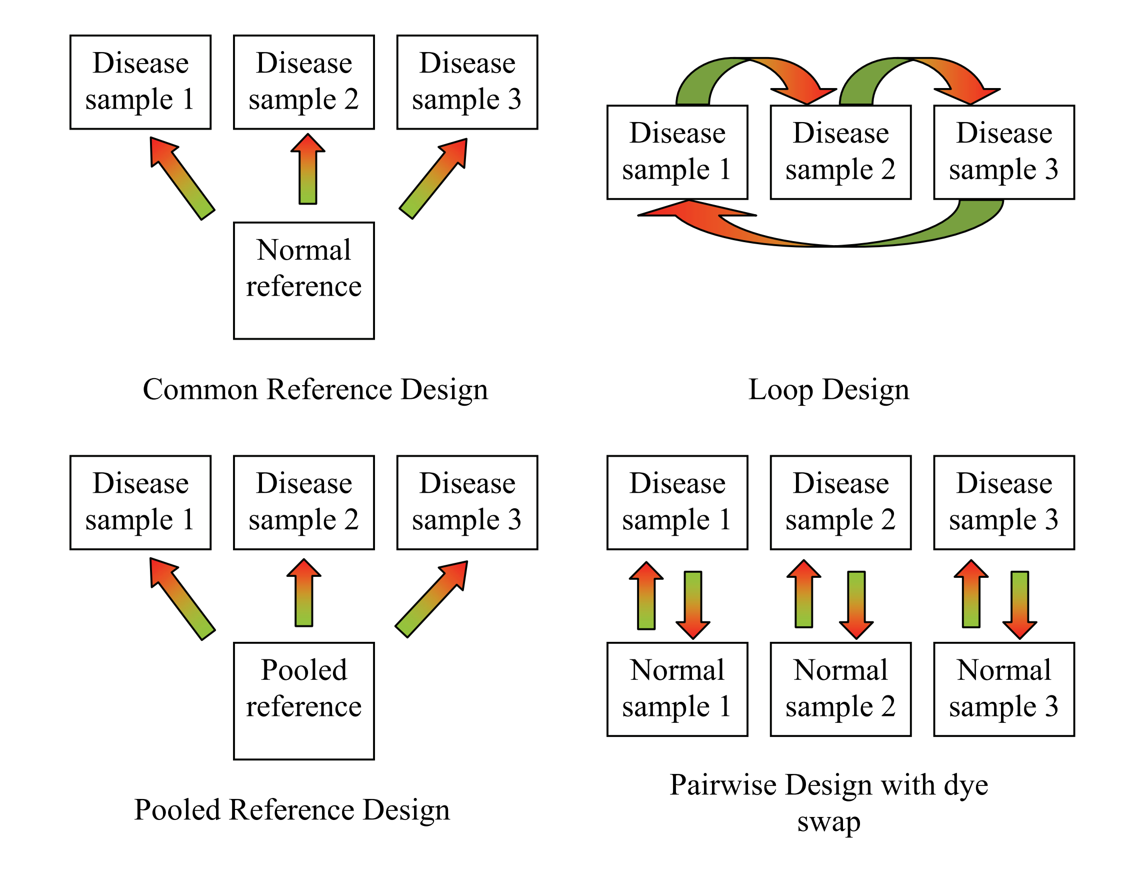


Figure 8: 2-channel experimental design examples. Boxes represent samples and arrows represent hybridizations. The tip of the arrays represents one dye (red) and the tail a different dye (green). Dye-swaps can be used to remove confounding dye effects as shown in the pairwise design. In this case, two microarrays are used to compare two samples.

3 Microarray data analysis

A possible outline of a typical microarray experiment (with emphasis on data analysis) could be:

1. Biological question
2. Experimental design
3. Microarray experiment
    a. RNA extraction
    b. RNA labeling
    c. Hybridization
    d. Scanning
4. Image analysis
    a. Segmentation
    b. Spot mapping
5. Expression quantification
6. Filtering
7. Normalization
8. Expression data analysis
9. Biological verification and interpretation

Optimizing the design based on the experimental objectives was discussed above, and is dealt with in the two first steps in this outline. Both have implications on all other steps as we shall see below.

3.1 Image Analysis

A microarray result is in its most primitive form a collection of intensity values with a two dimensional structure. For each *x,y* coordinate in a scanned area, there is an intensity $I_{x,y}$, and for arrays with more than one channel $I_{c,x,y}$ (for channel *c*). This can be structured and viewed just like an ordinary computer image. We refer to each reported signal ($I_{c,x,y}$) as a pixel. Image analysis methods are used to locate the structures in the scanned array using I as input and producing as output a specification of the identified features. The features should correspond to the probes printed on the array and their geometry (morphology) should be accurately extracted. Reporter areas (where probes are printed) on the array must be found, bounded and accurately mapped to a reporter list. The reason for the need of accuracy in this step is that the reporter signal must be separated from a potentially disturbing background signal and, even more important, it is crucial to know what probe is actually printed there. In literature these steps are known as segmentation [49] (or spot finding) and spot mapping. Many methods have been proposed and implemented to efficiently read microarray images with hundreds of megabytes of information. Examples of segmentation methods for spotted microarrays are fixed circle, adaptive circle, adaptive shape and histogram based segmentation. Fixed circle is a very simple method for printed DNA arrays where a circle with the same diameter as a spot is placed in such a way that the pixels inside the circle are as different as possible from the pixels outside the circle. As spots rarely are perfectly circular, background pixels are often treated as foreground pixels with this method. The adaptive circle technique reduces this problem by adjusting the circle diameter to exclude as much background as possible, but it is still inferior to the adaptive shape and histogram methods that use morphology and intensity distribution to separate foreground from background.

3.2 Expression Quantification

When the spots are found and separated from the background, the next step is to calculate an intensity signal for the spot. The foreground signal is normally a statistic (such as mean or median) based on the intensity values of all pixels determined to be inside the spot boundary (from the segmentation step). The correct method for calculating a "true" intensity is quite controversial, and most image analysis systems will report many different statistical measures and let the user choose the most appropriate. Some effects such as spot morphology and background intensity distribution could be used to determine the best way of segmentation and signal calculation. For instance, if most spots do not resemble circles, the fixed circle segmentation methods should be avoided.

Another challenge is how to handle background influence to the foreground signal. If the foreground distribution is mixed with the background distribution and the background is uneven across the slide, a background correction method should be used. Background influence may change across print locations (typically between blocks) or arrays (see Figure 9A). Some image analysis systems give advanced background measurements where local and global background is taken into account so that this can be corrected by simple subtraction (e.g. reported signal = foreground signal – estimated background signal).

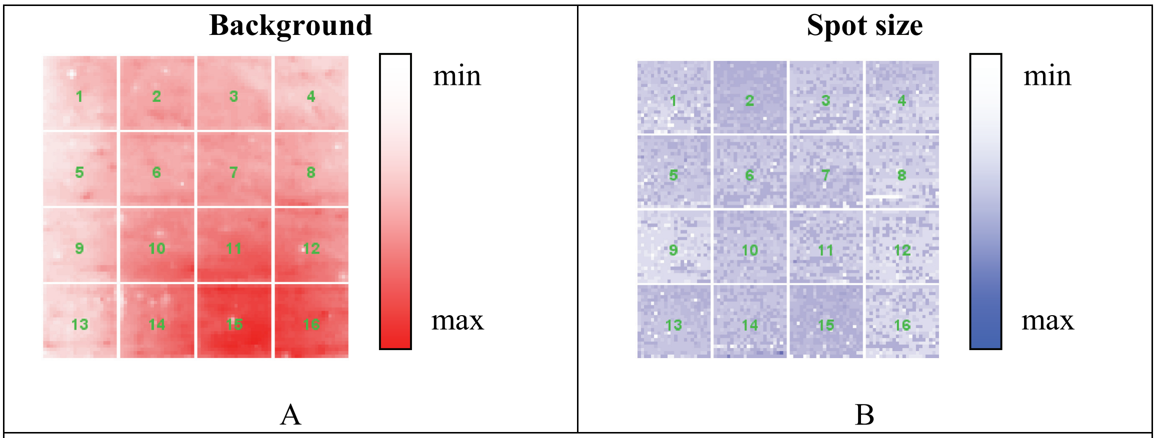


Figure 9: Two images generated by the quality control feature in J-Express Pro. Figures are generated from two-channel cDNA image quantitation data from the GenePix 1.4 software. A: Spatial distribution of background (log(median background(Channel1))). B: Spot location and pin-group (spots printed by the same pin, resulting in 4x4 squares in the Figure) together with spot size information. The intensity of each pixel corresponds to the size of a certain spot. Spots in pin-group 2 are for instance generally bigger than spots in block 9. This shows a dependency between spots size and print pin.

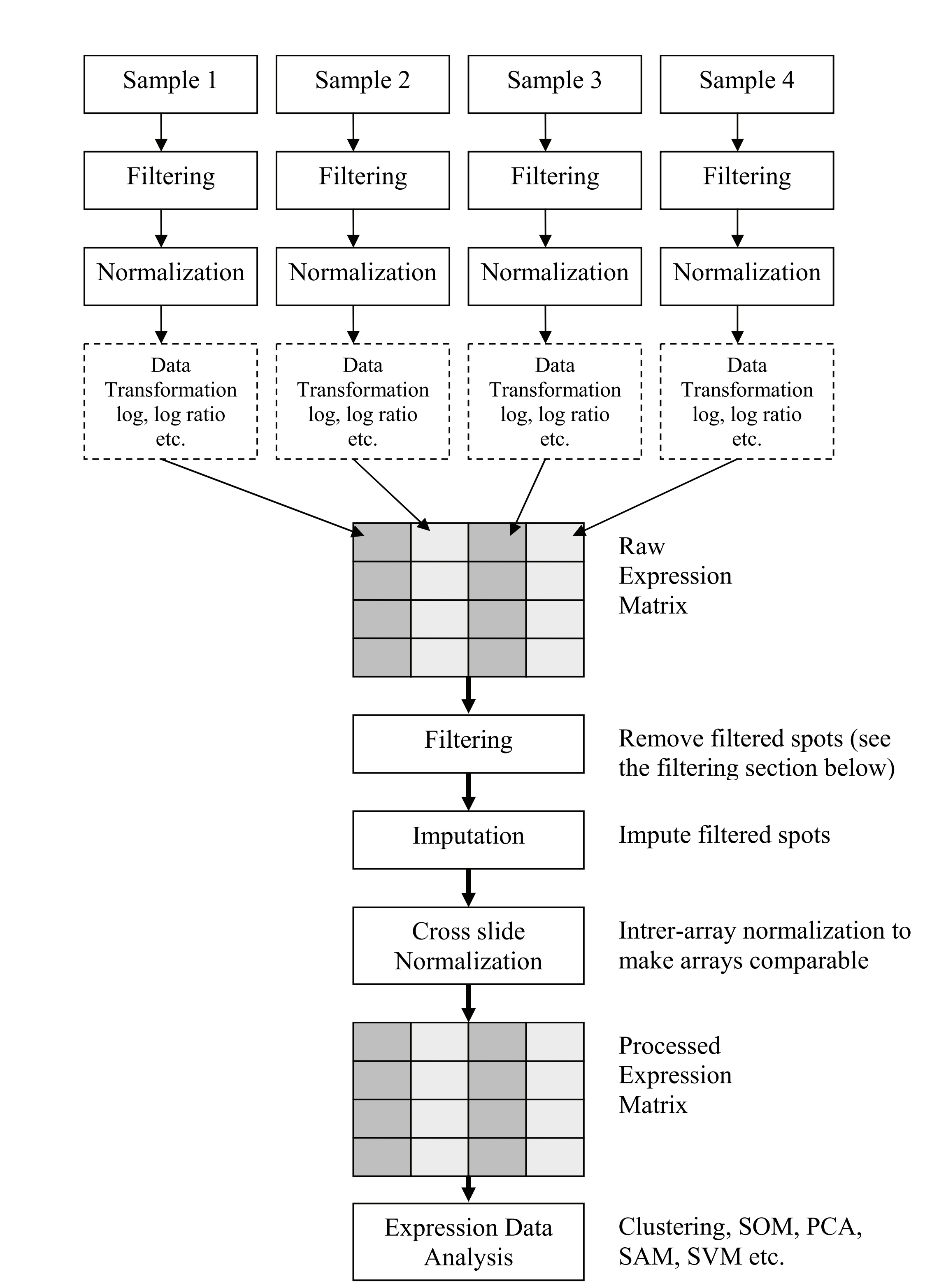


Figure 10: An example of a typical microarray data preparation process.

3.3 Filtering

In the image analysis step, some spots may be labeled as unreliable or unidentified by the software or by a user inspecting the data. The data from these spots should not be used in downstream analysis. Furthermore, it is customary to remove spots for which one suspects that the derived measures of (relative) hybridization will be unreliable. For example, if the intensity in the spot is not significantly stronger than the background signal, or if the variance of intensity in the spot area is too high, we may want to discard the spot in further analysis. Saturated spots are spots exceeding the range of values available to the scanning procedure. These can be corrected for by methods combining scans with different sensitivity (e.g. scans at different PMT voltages), but should be removed or tagged if uncorrected.

3.4 Normalization

As noted above, one of the objectives in the experimental design was to minimize the effects of unwanted biases. Still, some non-biological variation often ends up in the expression quantification data, and this should be corrected for whenever possible [50]. Normalization is the process where systematic bias from technical artifacts is reduced by making certain assumptions about the data. The aim is to adjust the data so that the resulting expression measurements are comparable in a probabilistic or statistical sense [27]. One assumption often made is that the total amount of transcription or the median/mean transcription in two or more measured conditions/samples is the same. Under this assumption, samples or channels can be scaled to an equal reference point (e.g mean, median or percentile) [51]. Such assumption must be valid in regards to an analogous biological assumption. For instance, if an assumption is made that the majority of genes do not change in expression between samples or that the distribution of gene expression values is the same, but the overall mean of the expression does change, such normalization can be fatal and will remove or reduce valuable biological information.

The normalization procedure should be tailored to the technology used. It should also reduce effects from all known sources of obscuring variation [52]. Some microarrays contain blocks with spots printed by the same print pin. The print pins may have physically different tips which may print morphologically different spots (see Figure 9B). The print tip variance can be removed by including this information in the normalization assumption (e.g. the mean log ratio signal within a block should be 0). Similarly, spatial normalization can use probe location to remove a spatial bias. However, by dividing the data to be normalized into groups based on location or print tip, the statistical quantity of observations decrease and the method often becomes vulnerable to overfitting.

If the experiment involves comparison of conditions/samples where one has reason to suspect that there are major changes in total transcriptional activity, other normalization methods such as those based on spiked in controls or constantly expressed genes (house keeping genes) can be used instead. Van de Peppel et al [53] have demonstrated how external controls and cell counting can be used in such experiments and how dramatically different the results are as compared to those obtained using the assumption about unchanged transcription level.

It is useful to visualize the relationship between pairs of intensity measurements (each from one array, or from one channel in a two-channel system) and log-ratios for a set of genes (probes). This is commonly done using MA plots (see Figure 11). It is not uncommon to observe an intensity dependent dye (channel) specific effect that shows up as a "banana shape" in these plots. There are normalization methods that correct for such effects, including spline [54] and local regression methods. The methods are based on the assumption that the total (or mean/median) expression level of most genes is unchanged also when one considers only genes falling within a local window along the intensity axis in an MA plot. The MA plot uses M as the y-axis and A as the x-axis where

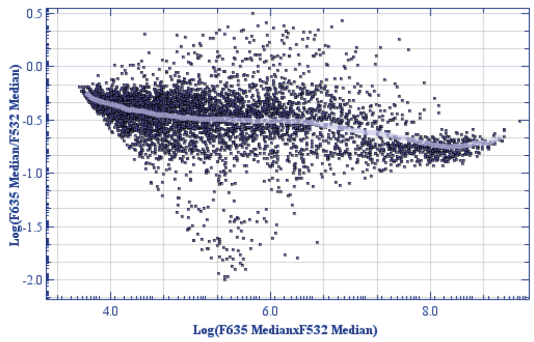
$$M = \log_2\left({R}/{G} \right)$$

and

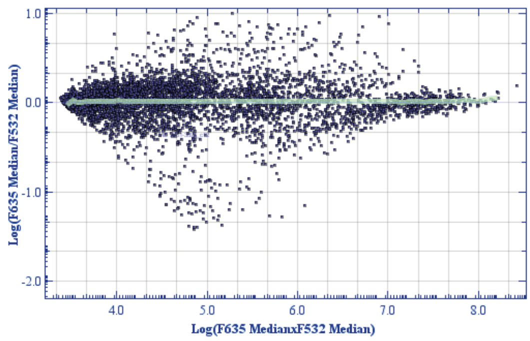
$$A = \log_2\left(R \times G\right) \text{ or } A = \log_{10}\left(R \times G\right) \text{ or } A = \frac{\log_2\left(R \times G\right)}{2}$$

The MA plot is sometimes referred to as an IR plot (I = intensity and R = ratio).

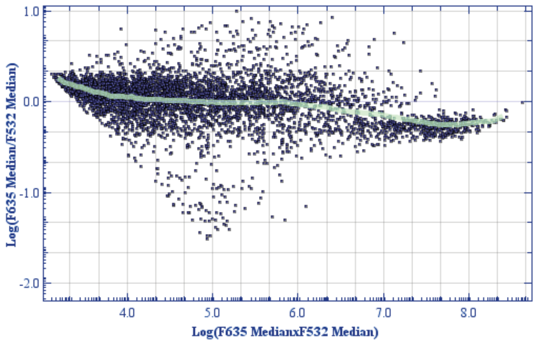
LOWESS [51, 55] (LOcally WEighted Scatter plot Smoothing) normalization is one of the most applied methods for normalizing two-channel arrays in the literature today (see Figure 11). Lowess removes intensity-dependent dye-specific effects from the features and can be easily adapted to include spatial and print tip information.



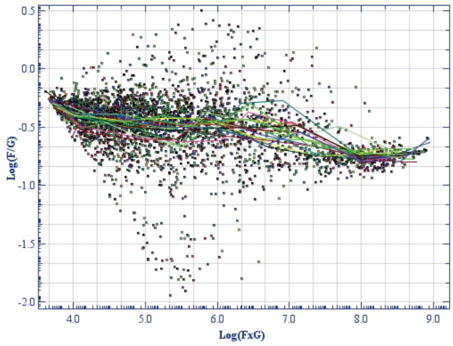
Un-normalized data



lowess normalized



Median normalized



Print-tip lowess regression curves

Figure 11: Some normalization methods visualized by MA plots (generated with J-Express). The median normalization shows the problem with linear normalization methods and although the median overall genes are unchanged, there is clearly some intensity dependent variation left. This variation is removed in the lowess normalization. In the print-tip lowess chart, the normalized data is not shown, only the source data with lowess normalization curves for each print tip. In the normalization result each of these curves will be straight, like in the upper right plot.

One-channel arrays are usually normalized in a slightly different way than two-channel arrays because there are no dye-specific effects such as dye/intensity as there are for two-channel arrays. Normalization in these arrays is often performed in regards to a single common reference (using methods similar to two-channel methods like lowess) or by making sure the distribution of intensities across all slides in the experiment is the same (e.g. quantile normalization [50], see Figure 12). The assumption made by the quantile normalization method is simply that the distribution of gene abundances is nearly the same in all samples.

When comparing signals from different dyes in a two-channel experiment or from the same dye from multiple arrays, the goal is usually to see if the signals are significantly different. We can refer to such analysis as looking for genes with a certain fold-change. Due to many of the unwanted biases described above (dye effects etc.) the intensity signals can generally not be directly compared, but must first be calibrated in some way like normalization. A simple fold change analysis can be to calculate a ratio or a log ratio between the two channels in a two-channel microarray. This is however a very simple analysis and it has been shown that the variance in many microarray studies is very intensity dependent [56], leading to an intensity-fold change dependency. If the variance is not normalized in a way that ensures the same variance across the whole intensity range, a fold change at intensity (R1,G1) will not be the same as a fold change at intensity (R2,G2). The process of normalizing the variance to create a more correct transformation is referred to as variance stabilization [57-59].

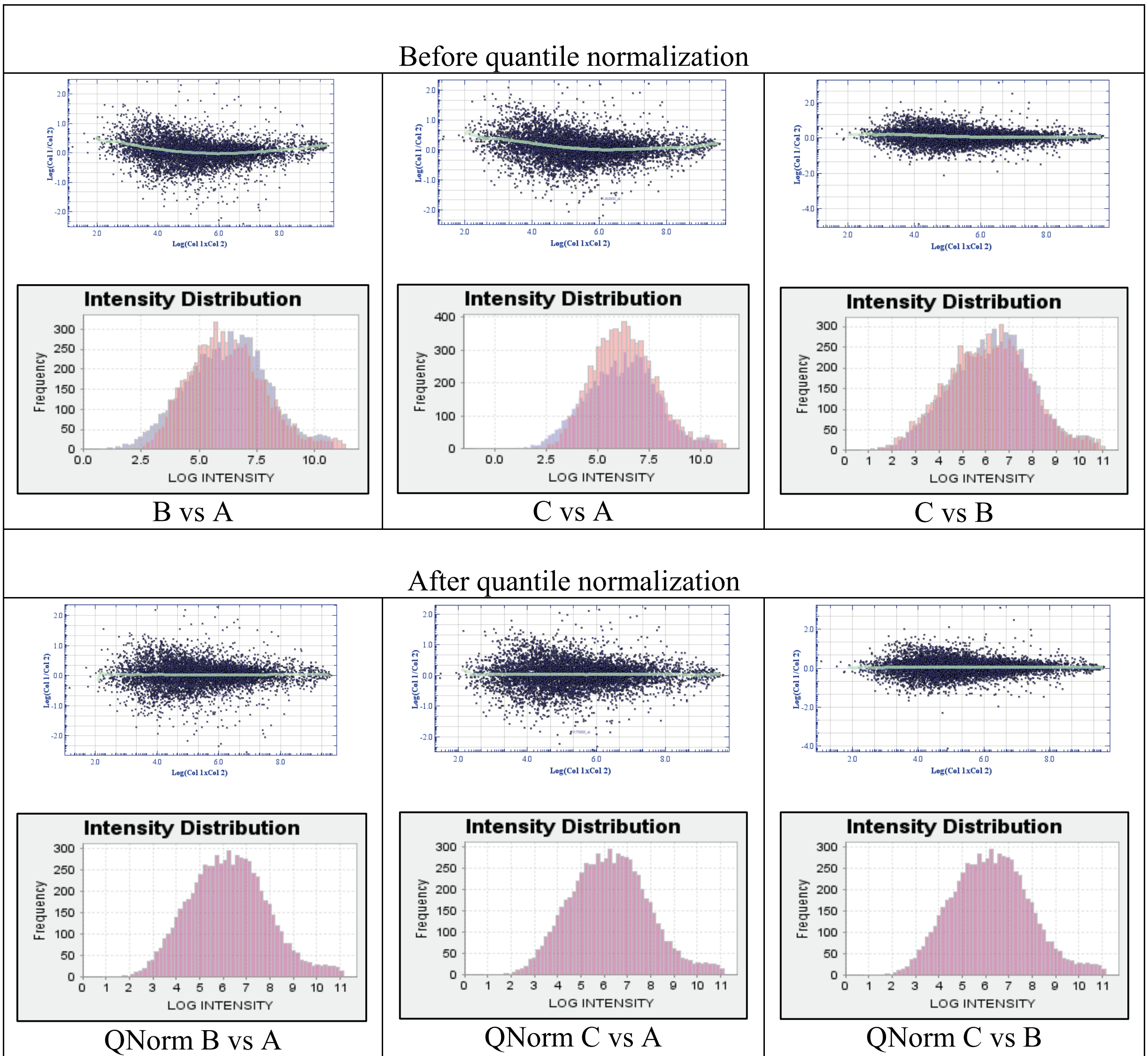


Figure 12: Quantile Normalization of 3 Affymetrix Genechip ® arrays visualized in J-Express. Top charts are MA plots with trend lines. Bottom charts are the corresponding histograms showing the intensity distribution.

3.5 Expression Data Analysis

After filtering and normalization, the microarray experiment can be organized into a gene expression matrix where each gene is represented by a row and each sample by a column. For each gene i there will be a certain number of samples j and the expression of a gene in a sample can be indexed as exp(i,j). For simplicity, both within-array replicates and technical (array) replicates are, when present, often merged into a more robust composite signal. Alternatively, the individual measurements can all be retained and consistency among replicates can be used to assess reliability in downstream expression analysis.

Gene/Time	T01	T05	T10	T15	T20	T25	T30	T35	T40	T45	T48
PFM0045	-1.5	-0.9	-0.7	0.4	0.5	0.5	0.3	-0.8	-2.0	-1.8	-1.4

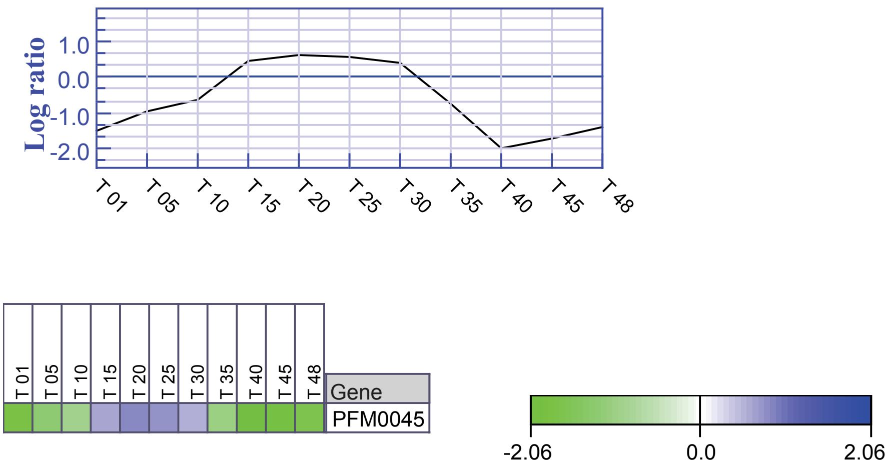


Figure 13: 3 different ways of visualizing the same gene expression profile.

After normalization there are often still unreliable values (usually those where the foreground signal is close to background) that should either be tagged with a quality measure or removed before expression analysis begins. Removing values (see filtering in section 3.3) will leave "holes" of missing values in the gene expression matrix if un-filtered replicates do not exist. If individual genes or arrays with a very high proportion of missing values exist, these are often removed entirely from downstream analysis. However, if only genes and arrays with no missing values are kept, much valuable data will be removed.

Some data analysis methods can handle missing values simply by not including them in calculations. However, many methods, such as those based on similarity measures, requires a complete expression vector (a valid exp(i,j) for any gene i and sample j). Similarity measures are used in many popular methods such as clustering and projection (e.g. multidimensional scaling). One of the most frequently used similarity measure is the

Euclidean distance metric where the distance from gene i to gene k given expression matrix x is given by the equation:

$$d_{ik} = \sqrt{(x_{i1} - x_{k1})^2 + (x_{i2} - x_{k2})^2 + \ldots + (x_{in} - x_{kn})^2}$$

If for instance, gene i is missing a value in sample 2 (x_{i2}) the calculation will fail or the distance will not be comparable to other distances in the data.

Missing values can be estimated with a data imputation method [60]. Imputation is a process where patterns in the expression matrix are utilized to predict missing elements. The assumption underlying imputation methods is that the missing value would (if present) follow patterns present in the data. For instance, if two genes are correlated and one of them has one expression value missing, the value would be such that the correlation is preserved also in this point. Different imputation methods are able to capture and utilize different types of patterns in the expression data. The simplest imputation methods are those replacing missing values with an average of non-missing values in the same row or column. This was often performed in early studies, but has been replaced by much more sophisticated methods such as the KNN method [61] and LSimpute [62].

The KNN imputation method is one of the most popular imputation methods in literature and estimates missing values by locating the *k* gene expression profiles that most resembles the one missing a value. A weighted average is then calculated from the k similar profiles and put into the expression matrix.

3.6 Gene Expression Analysis

The analysis of a gene expression matrix can be performed on data from the expression matrix alone, or by including prior knowledge such as gene function or sample disease state to discover patterns and relations in the data. These two fundamental types of analysis are referred to as unsupervised and supervised data analysis respectively [63].

Unsupervised analysis methods include self-organizing maps, hierarchical clustering, k-means clustering and principal component analysis (see Figure 14). Clustering methods [64] have been extensively used to analyze microarray analysis because of the way they reduce the data complexity by grouping together similarly expressed genes or samples. Interesting groups can then be further analyzed or interpreted by manual inspection.

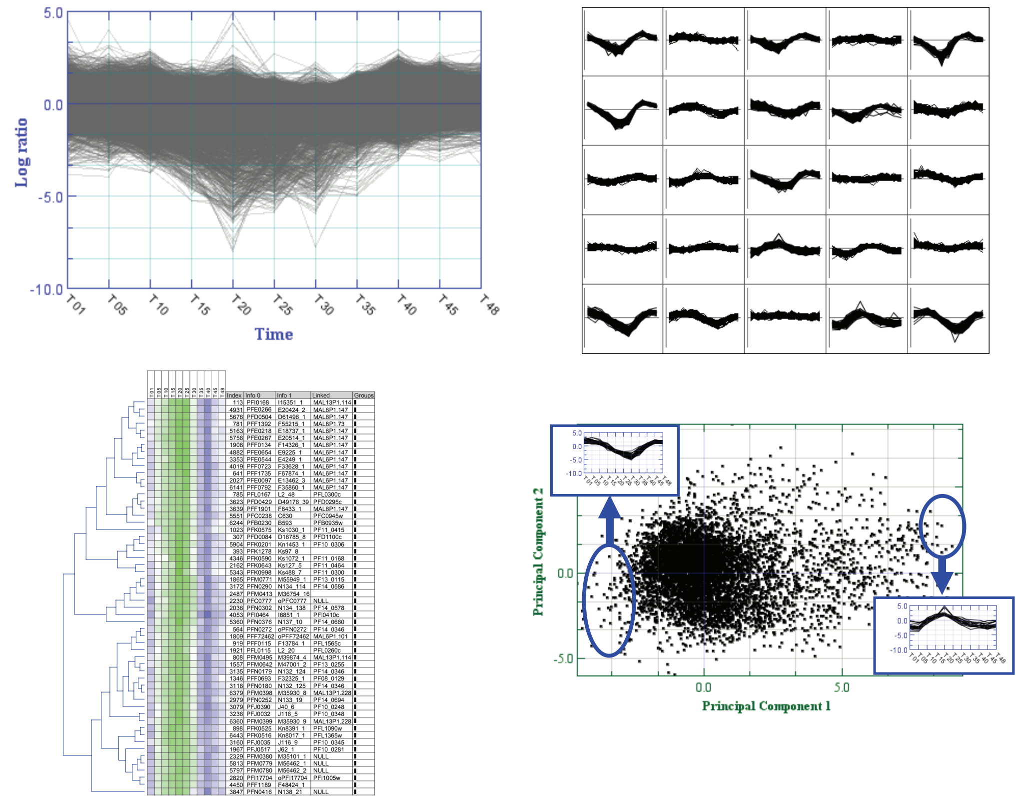


Figure 14: An unsupervised analysis example. Top left: Original time series data with one line per gene measured in 11 time points ($\log_2$ sample vs. time 0 ratios). Top right: The same dataset clustered with a k-means clustering (k = 25). Bottom left: A hierarchical clustering sub-tree of the top left data. Bottom right: A principal component analysis (PCA) plot of the top left data with two areas shown in line charts. All charts are produced with the J-Express software.

Supervised analysis is the other main group of analysis methods. With these methods, external information is utilized in search for coherence between patterns in the data and previously known properties such as sample class labels (diseased vs. normal) or gene groups (defined by for instance gene ontology terms or metabolic pathways). Examples of supervised analysis problem areas are classification [30] and methods for identification of genes differentially expressed between sample groups (e.g. SAM, see Figure 15).

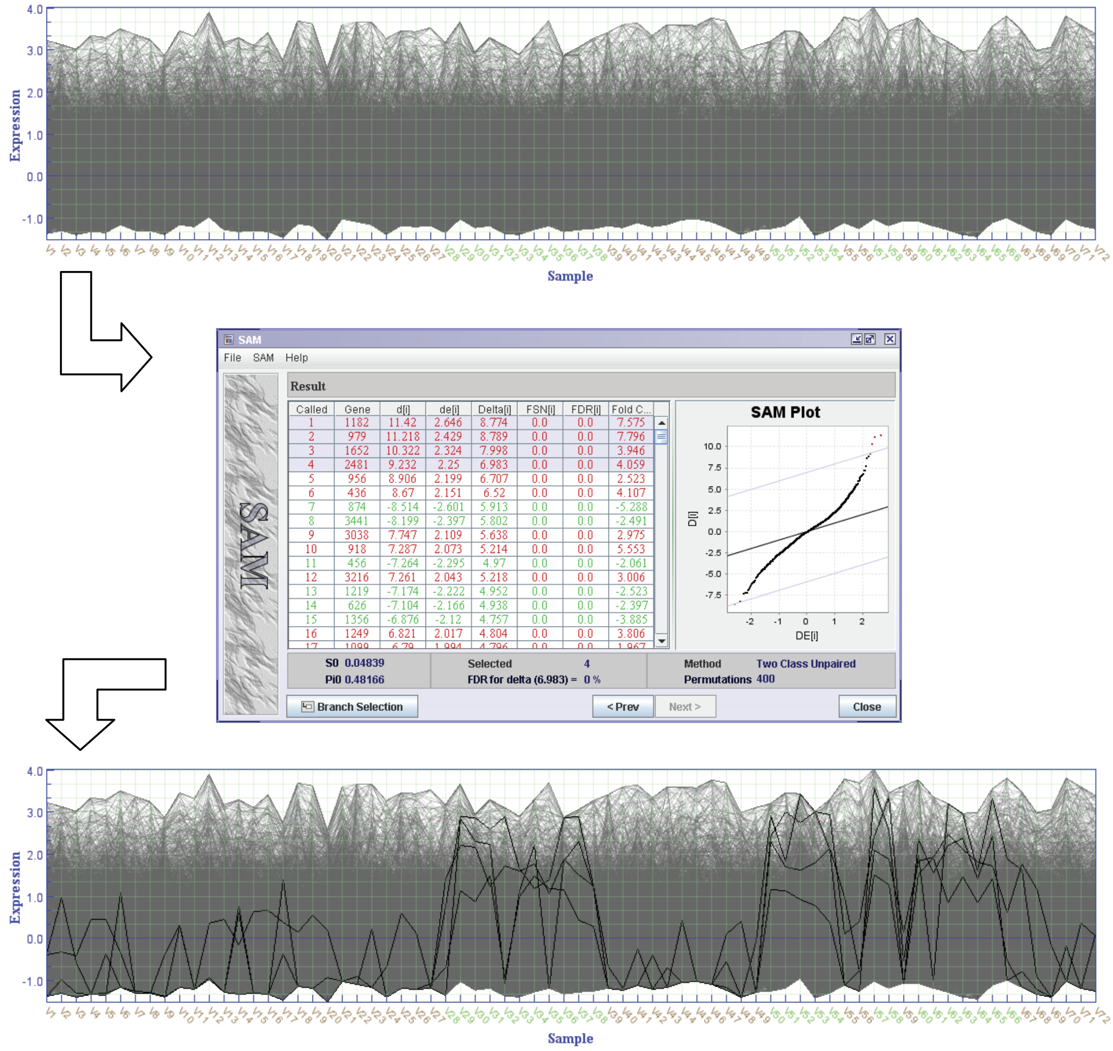


Figure 15: Supervised analysis example. Significance Analysis of Microarrays (SAM) [65]. Top: Original dataset with two different sample class labels (green and brown). Center: The SAM interface in J-Express. The four top genes are selected and shown in black (bottom). These genes are clearly expressed at different levels in the two groups (high in most green samples and low in most brown samples).

Identification of differentially expressed genes

From a gene expression matrix and sample class labels, a frequent goal for microarray analysis is to find genes differently expressed between the classes. For two-class studies, this can be done simply by comparing the mean expression in the two classes (see Figure 16). A more robust method is a t-test where the class variance is included in the equation and an even more robust method is SAM (Significance Analysis of Microarrays) where the variance parameter is controlled by ad-hoc statistics. Reducing the influence of variation in the statistics often give better results when handling microarray data. This is mostly because of the still rather inaccurate nature of microarray technology and the normally small number of samples in each group.

Feature selection and classification

Samples studied with microarrays can often be grouped into classes with certain common properties such as tissue location or cancer progression. When such properties exist, a common objective is to find genes differing in expression between the classes. The expression of these genes can then be compared to the expression of the same genes in new unstudied samples to determine class membership. Additionally, it can be interesting to see whether there are patterns in the expression data that are shared between some samples but not others. Such patterns may for instance suggest that a certain type of cancer has two different expression patterns which respond differently to treatment [26]. Many methods for finding sample groups based on expression patterns exist, and are referred to as methods for class discovery. Finding genes with classification properties is normally referred to as feature selection [66, 67] and using patterns from these genes to classify new samples are referred to as class prediction or classification.

Classification is normally performed in two steps: 1. Selecting features and 2. training a classifier rule [68]. For instance, a simple classification algorithm can "learn" that for two particular genes in a learning dataset with 20 normal samples and 20 disease samples, expression values are always high in normal samples and low in disease samples. For new samples, the algorithm (with the classifier rule) compares the expression values of these two genes to what it has learned and returns a prediction. The strength of the prediction in this example can be evaluated by studying the expression and variation of the learning set like a regular t-test (see Figure 16). Low within-class variations and high between group variations will most likely result in classifications with higher confidence.

The classification method is often specialized to handle features with certain properties. This means that not all feature selection methods are directly compatible with any classification method. Bø et. al [32] for instance, showed that feature sets selected based on pairs of features (genes) outperformed those based on individual genes for some datasets (see Figure 16). The discriminate power of these pairs does however not apply when the features are regarded separately, which is the case for many known classifier rules.

Feature selection does not necessarily mean extracting subsets of features. Many methods use weighting to include all features but differentiate their importance in regards to the classifier rule. For instance, an insignificant feature can be assigned a weight of 0.0, which generally corresponds to removing it from the list.

Two ways of selecting features for classification exists: 1. using the classification method in the feature selection procedure, and 2. separating the classification method and the feature selection method. The first is also known as the wrapping approach while the latter is known as the filter approach. A good thing about the wrapping approach is that features are optimized for the classification function which ensures optimal classification success with the used feature set. A filter approach on the other hand can use previously implemented methods for feature selection to create feature subsets for new classification methods. Compatibility between the feature set and the classification method is up to the user to verify.

The methods mentioned above for finding features differentially expressed between groups can also be used in a filter approach to select features with good discriminate power. Both t-test and SAM would create feature lists well suited for a LDA [69] based classification method (see below).

Feature subset selection is often referred to as being either "forward selection" or "backward elimination" when choosing the optimal number of features to use. Forward selection is a selection method where more and more features are added until a prediction rate stops increasing. Backward elimination is a similar method where the starting feature set is reduced until the prediction rate no longer increases.

One of the simplest classification methods is the K-Nearest Neighbor (KNN) method that simply counts occurrences of class labels for the K most similar samples and returns the most frequent. Support Vector Machines [20, 70, 71] (see Figure 16) is another supervised method that finds an optimal separating hyperplane between members of different classes in an abstract space, such as the multidimensional space of a microarray gene expression matrix. Other classification methods frequently used for microarray data analysis are decision trees [72] and Linear Discriminant Analysis (LDA) [69].

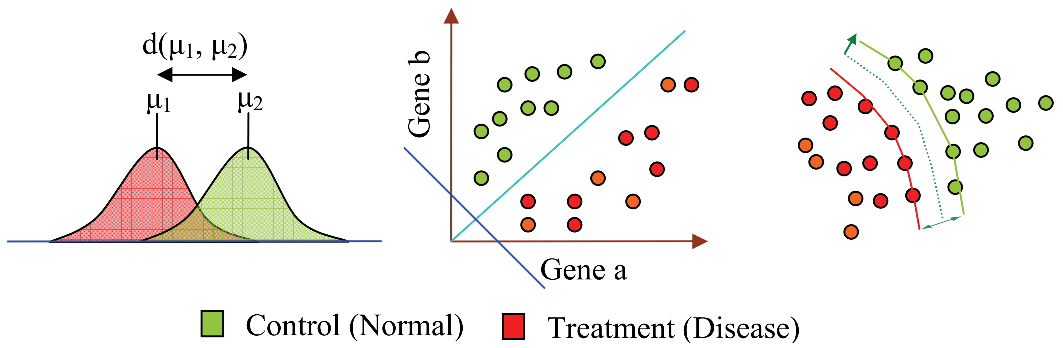


Figure 16: Feature selection and feature scoring. Left: Difference in expression mean between two classes. Middle: How pairs of genes can lead to better predictive power than each of them individually. Projecting the gene expression values (dots) on the blue line separates all samples perfectly. Right: SVM with a non-linear decision boundary.

Cross validation can be used to test the power of a classification method. Given a dataset containing pre-classified samples, the basic idea is to leave out a portion of the data (prediction set) and "train" a classification function from the rest (training set). The classification function is then used to predict the left out samples which are compared to the "correct" labels. For establishing a stable and reproducible success rate, the training set and prediction set should be non-overlapping and multiple tests should be balanced so that all samples are cross validated approximately the same number of times.

Cross validation is often referred to as leave-k-out cross validation where k corresponds to the portion or number of samples extracted as prediction set. For instance, a leave-one-out cross validation (or LOOCV) classifies each sample once using a classifier trained from all other samples. The good thing about leaving only one sample out is that the success rate can easily be compared to that of other classification methods. Comparison between methods where more than one sample is predicted either requires the list of left-out samples or a mean of multiple cross validations (preferably with some variation measurements). Leaving only one sample out in the cross validation does however often lead to overfitting of the prediction model. More robust cross validation methods leave out a greater portion of the data, such as a third of the samples.

Many methods used in supervised analysis of microarray data, like methods for creating feature lists and finding genes differentially expressed between classes, involves the testing of thousands of hypotheses. Controlling the increased number of type I errors (false positives) when testing microarray hypotheses (such as that a gene is differentially expressed between classes) is important because of the normally large number of genes that are measured simultaneously. The reason is that we are rarely interested in the intersection null hypothesis which is that all hypotheses are true or not, but instead the individual outcome of each single measurement. A very restrictive method for multiple testing correction is the Bonferroni procedure where the resulting p value is multiplied by the number of tests performed. In practice, a p-value cutoff at 0.05 in an experiment with 1000 genes is changed to 0.00005 with Bonferroni correction. A more popular, and less restrictive procedure is to include a False Discovery Rate (FDR) [73, 74] with the gene set and p-values reported. This value gives an average fraction of false discoveries when a decision rule is applied repeatedly to a set of hypotheses, but does not reduce p-values or the number of features reported as significant.

Permutation tests [30, 65] are often used to obtain a prediction score associated with feature selection methods and methods for finding genes differentially expressed between classes. This is done by repeatedly shuffling class labels and calculating new feature scores. The mean shuffled feature score is then compared to the unshuffled feature score. Permutation tests are especially efficient when the number of samples is sufficient to give a desired degree of confidence. Permutation methods also give more reliable correction for multiple testing.

3.7 Microarray result validation

A successful microarray experiment gives an answer to the biological question that was the starting point for the study. This may be a list of genes differentially expressed between two disease states, identification of new subtypes of a disease, or hypotheses about regulatory relationships. In most cases, results obtained using microarrays are validated using more accurate low-throughput methods such as quantitative real-time PCR, immunohistochemistry or northern blot hybridization.

Quantitative real-time PCR (polymerase chain reaction) is a method for logarithmic amplification of selected transcripts which are compared to standard controls. An approximate absolute copy number can then be established and compared to the result returned by the microarray experiment. The PCR procedure generally produces large amounts of gene copies by repeating a cycle of 3 steps: Denaturation (separating the two strands), annealing in the presence of primers (so that primers are attached to the single strands) and extention (rebuilding of new double strands). Because both strands are copied in each cycle, the increase of copies is exponential.

Immunohistochemistry is a method for identifying the presence of cellular or tissue constituents (antigens) by antigen-antibody interactions. The antibody binding is then identified either by direct labeling of the antibody, or by use of a secondary labeling method.

Northern blotting [14] involves the electrophoresis of the total mRNA for separation and blotting onto a membrane where a labeled probe is used for staining. It is a simple, frequently used method in molecular biology but has its weakness in that it requires relatively much mRNA.

Using external information in gene expression studies

As more and more analyses of molecular processes and regulatory systems are being completed, new methods including this knowledge in the data analysis are also emerging. For instance, patterns from unsupervised cluster analysis can be compared to metabolic and regulatory pathways or gene ontology groups to see if similar expression patterns may be caused by underlying biological gene regulation.

Semi-supervised clustering [75] is a relatively new approach to microarray analysis. In this type of analysis, the data is partitioned into classes based on supplied annotation and analyzed to find partitions with predominant patterns (or vice versa). The term semi-supervised comes from the combination of both un-supervised and supervised elements. A very simple form for semi-supervised analysis is simply to divide the data into groups corresponding to metabolic pathways or gene ontology groups and calculate a within-group gene expression correlation. By calculating an average background correlation from permutation tests (as described in the feature selection and classification section), pathways or ontology groups can be reported as relevant to the experiment objectives.

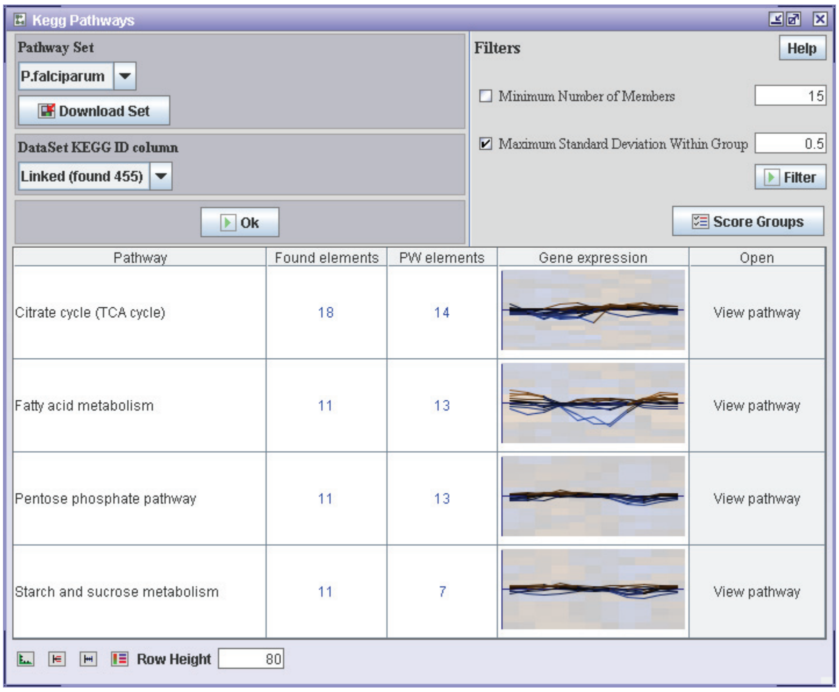


Figure 17: Pathway analysis in the J-Express software. The table shows metabolic pathways in the malaria parasite (*P. falciparum*) having a gene expression standard deviation (across 7 time samples) below 0.5. The table shows pathway name, number of elements found both in the dataset and pathway, number of elements in the pathways, gene expression chart and a button for viewing pathway details. Note that many-to-many relations exist and this is the reason why 18 elements (genes) in the dataset match one or more of the 14 elements (genes) in the Citrate Cycle.

4 Microarray data organization and storage

4.1 MGED and the Microarray Gene Expression (MAGE) standard

The Microarray Gene Expression Database Group (MGED http://www.mged.org) was initiated at Cambridge, UK in 1999 with the purpose of establishing recommendations for microarray data annotation. The group released MIAME 1.1 (Minimum Information About a Microarray Experiment [76]) in April 2002, which is a document describing a minimum of information that should follow a microarray dataset for the experiment to be reproducible. Besides the MIAME working group, several other groups exist within the MGED society. The MAGE [77] group is responsible for the MAGE (MicroArray Gene Expression) object model (MAGE-OM, see Figure 18) which is the conceptual model for implementations such as the array express database and the MAGE software toolkit (MAGEstk).

The reason for the MGED initiative was a great variety of new contributors to the microarray field, but no common language to describe the same “parts”. MGED was established in collaboration with many of the main industrial microarray suppliers such as Agilent, Axon and Affymetrix and have several main objectives:

- Promote adoption of standards (MIAME and MAGE) for DNA-array experiment annotation and data representation
- Assist development of academic and commercial MIAME and MAGE compatible software
- Develop microarray data quality standards
- Facilitate development and adoption of standards for describing high-throughput life science experiments (particularly microarray experiments)
- Continue development of MIAME and MAGE to include other types of microarray experiments.

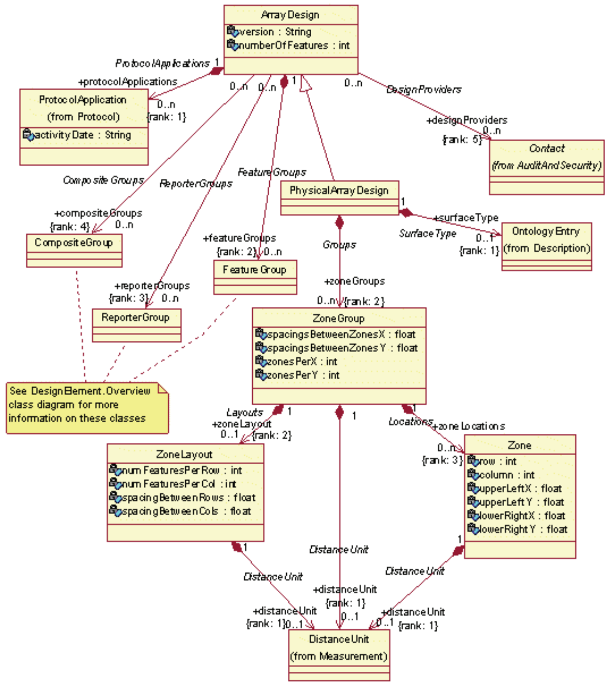


Figure 18: Example of a class diagram from MAGE. The ArrayDesign overview.

The MAGE object model also exists as an XML implementation called MAGE-ML (MAGE Markup Language). This makes it possible to encode any instance of the object model (containing actual data) as a sequential document to be sent over networks or exported to files.

Many popular journals now demand microarray experimental data to be available in curated databases such as the ArrayExpress database. One of the simplest ways for scientists to transport the data from their local systems to a MAGE compatible database is to locally export it to MAGE-ML format using the MAGEstk toolkit (or a MAGEstk based software tool). The MAGE-ML package can then be sent to curators who will make sure it contains the information necessary and submit it to the database, where it can be made publicly available trough internet servers.

Although the introduction of MAGE has made storage and retrieval of microarray data and experiment information easier, it still has some limitations. The MAGE object model is implemented in a way that maximizes compatibility with new structures and thereby increases the complexity. MAGE-ML output files are very complex and almost impossible to understand without a MAGE-ML reader that can rebuild the object structure. In addition, the toolkits for creating MAGE models lacked a structure for representing expression analysis in an effective manner. Our group has however added this feature to the MAGE software toolkit and implemented support for storing downstream data analysis [78] (see Figure 19). This addition has been made available in the J-Express software.

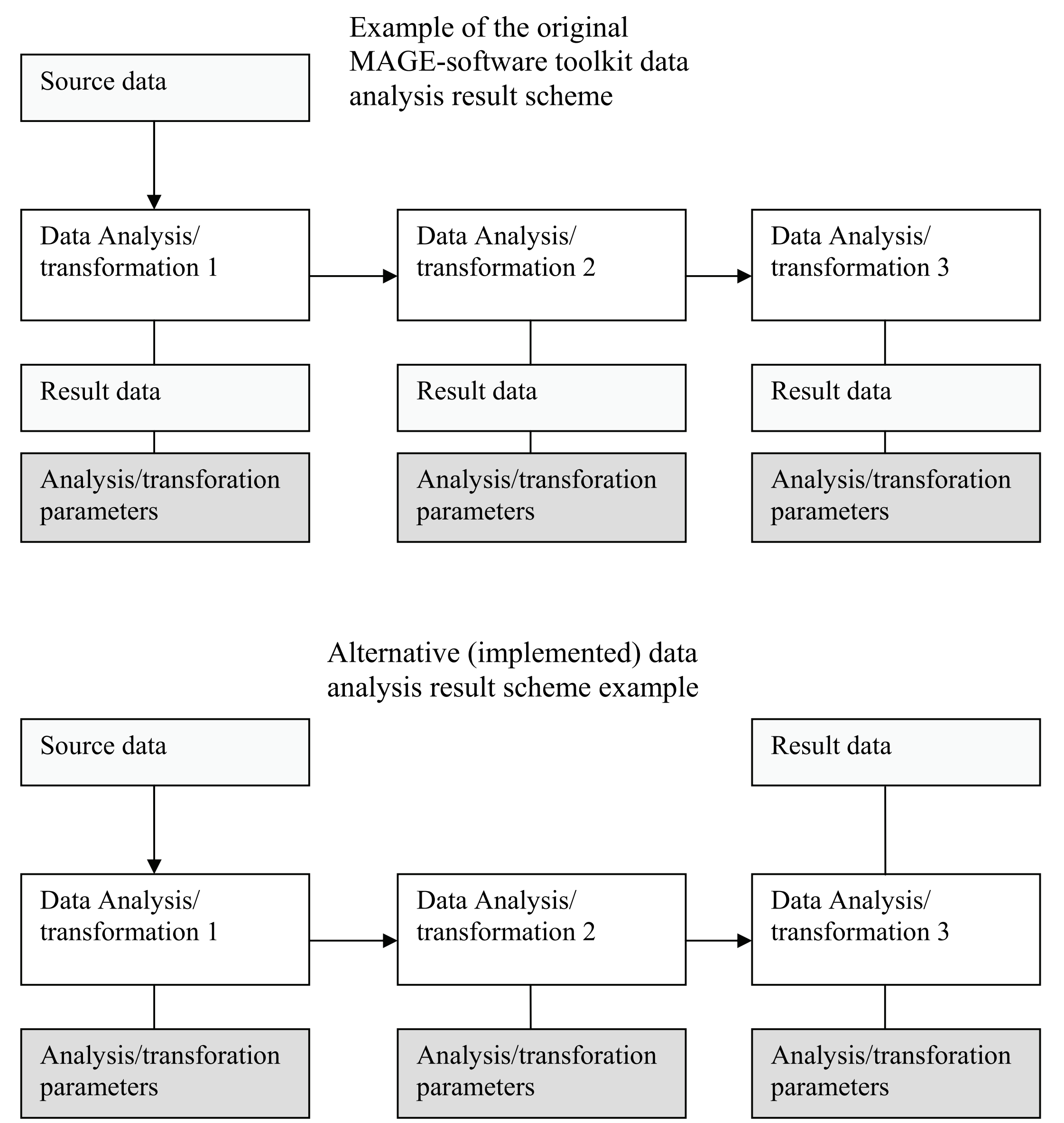


Figure 19: Two ways of storing microarray expression data analysis results using the MAGE software toolkits. Top: Original scheme for analysis organization with each analysis block containing input and output data. Bottom: Our contribution to the toolkits exemplified by a simple example. Instead of storing all in-between data, only the source and final data is stored together with all information needed to recreate the full analysis.

The framework for pre-processing of data from image-analysis systems is flexible and allows for scripting and plugin of new novel methods. Import filters for most file formats are included in the software and for new unknown formats new parsers can be built relatively easy.

Low-level analysis and data preparation often ends with the compilation of a gene expression matrix. This is handled effectively by the J-Express software in a comprehensible manner. The expression matrix can be put through a replicate combination tool, a missing value imputation tool and a data transformation tool (intensity values to ratios, log ratios etc.) and finally end up in a project tree. This is the beginning of the next analysis phase: downstream analysis. Data analysis methods come in two versions in J-Express. One collection of tools is the unsupervised methods such as clustering and projections. Both genes and samples may be tagged with group information such as functional gene group, metabolic pathway, sample disease state or sample location. This information can then be used directly in the unsupervised analysis to evaluate results in a semi-supervised fashion. Group information can also be used directly in the supervised methods such as t-tests and SAM. Pathway and Gene Ontology analysis tools can be used to discover over-representation of genes with special expression patterns in functional groups or metabolic patterns.

The hierarchical structure of the data analysis lets the user start with a complete gene expression matrix and extract sub-datasets for more specific studies. This enables a fast and effective analysis, using only a fraction of the resources a full-set analysis would require. A scripting interface in the downstream analysis makes it possible to automate repetitive tasks and plug in new in-house implemented methods for data analysis and storage. Scripting is handled by a Jython interface (http://www.jython.org) which is an implementation of the popular python language in java, as well as providing scripting access to all java libraries (including standard libraries, downloaded libraries such as those for scientific and statistical computing and J-Express libraries).

Many methods for data management and transformation exist J-Express. Some important tools are the annotation manager which can merge annotation information (such as database accession numbers and gene description), and the filtering tool for removing uninteresting genes based on properties like missing values and expression variation.

Some external net-resources are also available in J-Express. One net-feature is the automatic download of Affymetrix CDF (Chip Definition File) files when only CEL-files are available. Other net resources include a server-side dataset repository for sharing datasets and automatic installation of software updates.

The charts and graphical results produced by J-Express can be exported to many different image-formats. Importantly, J-Express can export vector graphics (in contrast to bitmap graphics) which greatly improves quality both on-screen and printed versions.

The J-Express interface is generally easy to use and is designed in such a way that it should be easy to apply various methods to the data. The flexibility can however lead to unwanted processing. For instance, in the low level data preparation component, one can add filters and normalization methods (called processes) in a table. The process at the top of the table will be applied first and the bottom process will be applied last. If a normalization process is placed at the top and a filter process below, this will lead to a normalization based on unfiltered data (including control spots, empty spots, flagged spots etc.) which is not necessarily a good way to perform normalization. By putting the filter process above the normalization process, the normalization is based only on good spots and has a better basis for making necessary assumptions (see Figure 20).

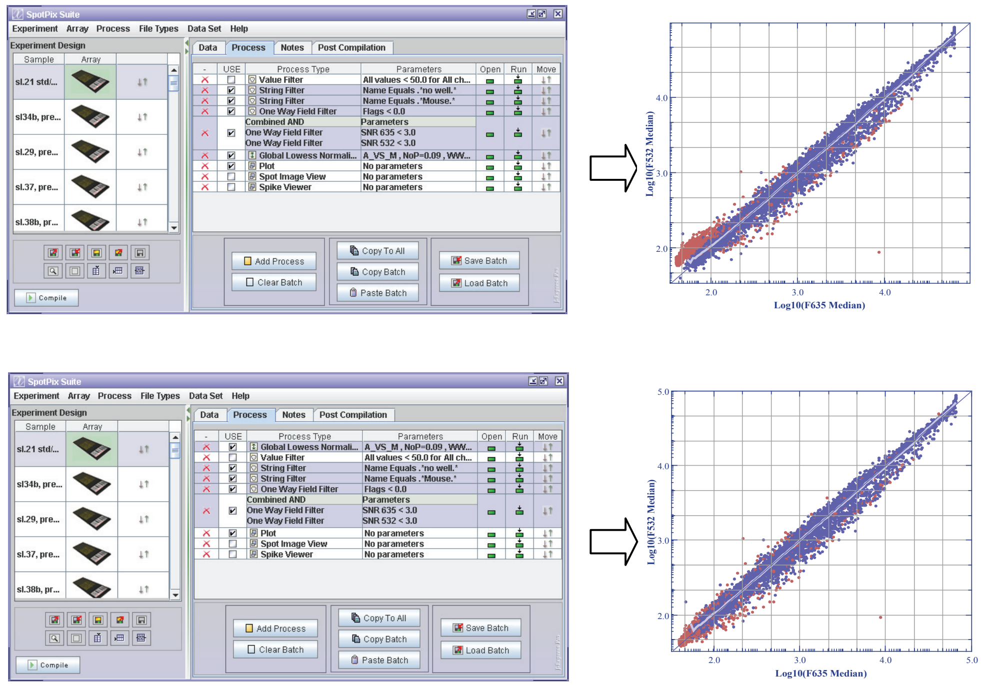


Figure 20: The preprocessing framework in J-Express shown here with two pairs of process stacks and log-log plots (see Figure 7 for information about the log-log plots). The left windows show the preprocessing functionality with microarrays and a process stack. Each row in the stack is referred to as a process and can alter or display the data. When executed, the processes will be applied to the data from top to bottom if not disabled (by un-checking the USE box). The right charts show the plot process in the process stack after the blue processes above it have been applied to the data. The top process window and log-log plot shows a process with some filter steps first, then a global lowess normalization process. This leaves all filtered (red) spots in the right log-log chart unchanged (and neglected by the normalization method). In the bottom process window and log-log plot the normalization process has been moved to the top of the stack, resulting in normalization before filtering.

It is always possible to inspect the complete line of data-transformations and analysis steps performed in a J-Express analysis. This is handled in the J-Express project tree where each data-transformation or analysis step is added as a new dataset and linked to the source (as a child node in a graph). Each new dataset also contains a list of all methods (and method parameters) performed from the root dataset to the leaf. This makes it possible to recreate any analysis, and to go back and redo steps with different parameters when desirable.

Many ways of controlling the data and analysis quality are also integrated in J-Express. This is necessary because an analysis method performing well on one dataset does not necessarily give good results on any dataset. For instance, when normalizing data it can be of great value to see how the normalized data is distributed so that the parameters can be changed if the results are bad (see Figure 20).

J-Express version 2.7 contains a codebase with over 220.000 lines of java code. Disregarding a few percent programmed by collaborators and some open source elements, most has been programmed during the master and Ph.D projects of the thesis author. Besides this codebase, J-Express also uses many precompiled public libraries such as JFreeChart for some of the graphics, JSCI (http://jsci.sourceforge.net) for math and statistics and JAMA (http://math.nist.gov/javanumerics/jama) for matrix calculations.

6 Summary of papers

Manuscript 1:
Trond Hellem Bø, Bjarte Dysvik and Inge Jonassen**: LSimpute: Accurate estimation of missing values in microarray data with least squares methods. Nucleic Acids Research, 2004, Vol. 32, No. 3 e34**

Missing elements in gene expression data is a frequently occurring problem. As the microarrays that constitute the expression matrix are processed to ensure data quality, a varying number of spots are usually considered less trustworthy because of artifacts such as weak foreground signals (close to-, or mixed with background), dust, scratches and saturation. Some analysis methods can disregard these spots and some methods even make use of confidence levels in calculation, but in many cases a complete matrix is required.

We have proposed a method for imputing missing data in a gene expression matrix based on least squares regression. The success rate for this method is compared to one of the mostly applied missing value imputation methods, KNN impute and found superior for three public datasets.

LSimpute estimates missing values in the dataset by using existing information from the most correlated genes and samples. LSimpute_gene is a sub-method that calculates a result based on regression from the 10 most correlated genes. Similarly, LSimpute_array calculates a result based on regression from the most correlated arrays. We found that these two methods complement each other and often yield a better estimate when combined than individually.

We also implemented two methods for imputation based on expectation maximization; EMimpute_gene and EMimpute_array. These two are similar to the regression based estimation methods LSimpute_gene and LSimpute_array, but use a maximum likelihood estimate of the covariance matrix instead of the empirical covariance matrix. Both of these methods, EMimpute_array particularly, were able to compete with the least squares based methods on the test datasets.

Manuscript 2:
Bjarte Dysvik, Endre N. Vasstrand, Roger Løvlie, Osman A-Aziz Elgindi, Kenneth W. Kross, Hans J. Aarstad, Anne Chr. Johannessen, Inge Jonassen, and Salah O. Ibrahim: **Gene Expression Profile of Head and Neck Carcinomas Between Patients from Sudan and Norway Reveals Common Biology Regardless of Differences Related to Ethnicity and Life Style. Clinical Cancer Research. 2006;12:1109-20.**

The objective for this work was to look for gene expression differences between disease tissue and normal tissue in head and neck samples and to see if there were any consistent differences between patients from Sudan and patients from Norway. Because samples were gathered in different environments and from different locations in head and neck, the experimental design was one of the most challenging parts of this study. We chose to use several independent experimental designs and perform a meta analysis on the results.

A total of 8 separate gene expression studies from 72 Norwegian and 45 Sudanese samples were performed using 7 pairwise disease vs. normal designs and 1 disease and normal vs. a common reference design. The individual analysis of the 8 groups revealed 8 lists of interesting genes with lengths ranging from 35 to 578 candidate genes. We then searched for genes overrepresented between these lists by counting in how many lists each gene was found. The scores were compared to a similar study with 100 random generated lists of the same lengths as the candidate lists. This revealed a statistical significance of the top scoring genes. These genes were then subjected to further studies such as search for overrepresentation in metabolic pathways and gene ontology groups.

The results were also compared to genes found differentially expressed in 9 other similar studies as well as the HNCGAP (Head and Neck Cancer Genome Anatomy Project) database. Some patterns were found and suggest that habits such as snuff use which are only common in the Sudanese sample group have implications to HNSCC development.

Manuscript 3:
Bjarte Dysvik, Trond Hellem Bø and Inge Jonassen: **Mixture models applied to microarray gene expression data**

Feature selection and class prediction are frequent tasks in microarray data analysis. Feature selection is the process of selecting a subset of genes possessing a predictive property. This subset can then be used by a classifier method to "learn" a class dividing pattern which can be used to predict class membership of new unstudied samples. Many feature selection methods exists such as t-tests, SAM, and 1-way ANOVA, and many classifier methods exist, such as Linear Discriminant analysis (LDA) and Support Vector Machines (SVM).

In this project, we propose a new method for feature selection and classification based on mixture models. The mixture models are composed of n_j normal distributions for each group, where n_j is the number of measurements for group j. The motivation for this approach is the assumption of a not necessarily normal distribution of sample measurements and with that a better result with more flexible models. The flexibility is provided by the mixture model which (by modifying the standard deviation for the underlying normal distributions) can vary from strongly overfitted (low model standard deviations) to a normal density distribution. We suggest that somewhere in between these two values is a shape flexible enough to adapt to more than one single distribution while rigid enough to avoid overfitting.

We used two ways of estimating the success rate of the method (leave one out cross validation and leave one third out cross validation) and compared the results from 6

5 The J-Express software

J-Express is a software tool for analysis, organization and visualization of microarray data. Since its beginning in 1998, the software has evolved from being a simple collection of a few popular methods for expression analysis to a relatively advanced framework for low-level data preparation, expression analysis and data storage. As one of the first software providers we have integrated MAGE support both for data import and export. The MAGE export includes information about algorithm parameters and software settings. This information is automatically logged as data analysis proceeds, normally from normalization and filtering of single arrays to advanced analysis methods and often finally to the discovery of one or more gene-lists with interesting genes and their expression values. We have been working directly on the MAGE software toolkits to enable effective transportation and storage of microarray analysis results, which normally consists of large amounts of data, while still retain the information necessary to reproduce the results. One of the implemented solutions was to store only the data going into the first step of analysis and the data coming out of the last step (see Figure 19). In addition, we stored a description of the software package used, including a description, parameters and version number of all methods used in the analysis. This should be everything needed to recreate the results based on the same input. In addition, it reduced the size of the data package dramatically by leaving out intermediate results.

publicly available datasets. The results from all test datasets were promising and show that this method can compete and event outperform many of the most popular methods for feature selection and class prediction used today.

Manuscript 4:
Inge Jonassen and Bjarte Dysvik: **Current Protocols in Bioinformatics. John Wiley & Sons, Inc. Chapter 7 Analyzing Expression Patterns, Unit 7.3 Analysis of Gene-Expression Data Using J-Express.**

This chapter in the collection of protocols in bioinformatics describes a stream-lined execution of certain expression analysis tasks using the J-Express software tool. Data analysis is generally organized into 3 different steps; importing image quanitation data, preprocessing array data and downstream analysis of a gene expression matrix (the first and second step can be bypassed by loading a tab-delimited pre-processed gene expression matrix). The first step involves using a framework for importing image quantitation data from different image analysis systems. The second step describes a preparation pipeline for generating a gene expression matrix from many image quantitation files. This includes filtering, normalization, data transformation and all processing needed on each array to enhance data quality and make data from each array comparable to data from other arrays. The third step involves analysis of a gene expression matrix. Many methods for expression matrix analysis are presented in the protocol, including profile searching, clustering and projection. A synthetic sample dataset is used to show differences between some of the implemented methods in a comprehensible way.

Important concepts in this description of the J-Express software are those of a dataset and meta data. Each collection of data in the form of an expression matrix and annotation (on genes and samples) is stored in an object referred to as a dataset. Also included in this dataset is group information about genes and samples and a list containing meta data. Whenever a subset of genes or samples is extracted from the dataset or the data is altered in any way, a new dataset object is created and added as a child node. The child node meta list is then supplied with a new method description with detailed information about the process leading from the original dataset to the child dataset. Thus, a data analysis project using J-Express can be organized in the form of a tree with the original gene expression matrix at the root and branches with transformed and extracted data in branches and leaves. From any dataset node, one can view the meta list and re-create the whole analysis given the root dataset.

Manuscript 5:
Mai Lill Suhr, Bjarte Dysvik, Ove Bruland, Saman Warnakulasauriya, Asoka N. Amaratunga, Inge Jonassen, Endre N. Vasstrand and Salah O. Ibrahim: Gene Expression Profile of Oral Squamous Cell Carcinomas from Sri Lankan Betel Quid Users

Oral squamous cell carcinoma (OSCC) is a major health problem in Southeast Asia. It is believed that the widespread use of betel quid (a chewed mixture of areca nut, lime and catchu wrapped in a betel leaf, often with tobacco added) in this area is in some way related to the high occurrence of this disease. In this project, we collected tumor and corresponding normal tissue from 15 patients from Sri Lanka with OSCC diagnosis. RNA from each of these was extracted, labeled and hybridized to microarrays created by the Norwegian microarray consortium. After scanning and image analysis, data processing and analysis revealed 263 candidate genes found differentially expressed between normal and tumor samples. Further investigation of these genes using Gene Ontology groups and KEGG pathway information revealed significant overlap between the candidate genes and ontology terms and KEGG pathways. When searching for previously reported genes in relation to head and neck cancer, we found 66 of the 263 genes previously reported. Hierarchical clustering of the tissues using the 263 differentially expressed genes also revealed correlation between samples and factors such as tumor grade and size.

From an informatics and statistical point of view this project presented us with some serious challenges, much because the experimental design and hybridizations were conducted prior to the involvement of our department. The biologists chose a pairwise tumor vs. normal hybridization design with dye completely confounding with tumor state and without including dye-swap hybridizations. This generally made deciding a gene to be differentially expressed due to tumor progression impossible, without estimating the dye-gene specific effects. These effects were approximated in two ways. First, we excluded all genes reported to have dye effects by the microarray manufacturer. Second, an experiment using different tumor and normal samples, but the same microarrays and protocols were prepared. This experiment included dye-swaps for all sample pairs. When a gene was found significantly differentially expressed in our dataset (using a regularized t-score), the same gene was controlled for dye-specific effect in the dye-swap dataset. Although this procedure did not give us a 100% reliable dataset, any dye-effect still present should not contribute to analyses such as those including Gene Ontology or KEGG pathway information (under the assumption that dye effects should not be dependent on biological functions).

By further manual analysis of the 263 candidate genes, we found many genes of special interest for further studies. These includes genes previously reported as being correlated with tumor progression and some unreported genes involved in relevant biological processes such as angiogenesis and metastasis.

Manuscript 6:
Christiane Moros, Petter Frost, Bjarte Dysvik, Bjørn Kristiansen, Inge Jonassen and Frank Nilsen: **Identification of transcripts regulated during sexual maturation and egg production in adult female salmon lice *Lepeophtheirus salmonis* (*Crustacea*, Copopoda), using EST-sequencing and microarray analysis**

In this project, the goal was to study the maturation of the female salmon louse. Salmon louse (*Lepeophtheirus salmonis*) is a severe problem for the Norwegian aquaculture industry. We created a cDNA microarray by printing clones from a salmon louse EST library onto a microarray with assistance from the Norwegian Microarray Consortium. These microarrays were used to measure the relative expression of mRNA from 35 salmon lice divided into 7 development stages (a pre adult stage and stages T1 – T6 by visual inspection). We used reference hybridization design with a common reference created from a mixture of pooled mRNAs from different developmental stages and sexes. After data preparation and quality control, we compiled a gene expression matrix and grouped together lice from the same development stage. Gene targets in our downstream analysis were genes expressed differentially between the development stages. We created a group-wise expression matrix by combining expression values for all samples in the same developmental group using a trimmed mean. Prior to creating this matrix, we verified a high expression similarity among lice in the same group by methods such as principal component analysis and Self-Organizing Maps (SOMs).

Simple cluster analysis (SOM) revealed two expression patterns differing from the global none-changed pattern. One of the patterns were high expressed in the early stages of development but decreased in the late stages. The other pattern showed a low expression in the early stages and up-regulation during maturation. Further analysis involving both ESTs and samples in a bi-plot (correspondence analysis) also revealed a third pattern, similar to the first pattern, but with a high expression in the pre adult stage and very low expression in the latter stages. Genes with known function in the two first groups showed as expected growth related genes in the first pattern (being down-regulated) and reproduction genes in the second pattern (being up-regulated). Within these groups were also many novel genes.

Other publications:

Bjarte Dysvik and Inge Jonassen . **J-Express: exploring gene expression data using Java. Bioinformatics**, 17, 369-370 (2001)

Ayodele A. Alaiya, Bo Franzén, Anders Hagman, Bjarte Dysvik, Uwe J. Roblick, Susanne Becker, Birgitta Moberger, Gert Auer, Stig Linder. **Molecular classification of borderline ovarian tumors using hierarchical cluster analysis of protein expression profiles.** International Journal of Cancer, Volume 98, Issue 6, 2002, 895 - 899

7 Further work

The field of microarray data analysis is beginning to "grow up", at least compared to some of the methods used in the first publications. In addition, the underlying microarray technology, including labeling technology, hybridization techniques, microarray production, storage and scanners have improved dramatically the last couple of years. Prices have been reduced, enabling researchers to use microarrays more frequently, and in greater number. In addition, microarrays are now packed with more and smaller spots, producing considerable more data than before. More knowledge about genes and genomes are stored in public databases, and the number of sequenced organisms continue to grow. All these things are constantly affecting how microarray data analysis is performed.

The size of a microarray experiment is now often hundreds of samples with as much as 40.000 measurements. Compared to some of the first datasets, such as those containing abundance of 6124 yeast genes in 15-20 time points (e.g. Spellmans based cell cycle experiment using elutriation [79]) these datasets are often too large for efficient analysis using the same conventional algorithms. Although both computer power and physical memory size has increased together with the data size, some methods are still increasing exponentially in efficiency and time usage. An example is the frequently used hierarchical clustering method which takes at least a factor $O(n^2)$ time, and some implementations still takes $O(n^3)$ time and $O(n^2)$ space. Using an $O(n^3)$ algorithm means that doubling the sample size increases time usage by an eightfold, in a best case scenario. For such algorithms, we often need to reduce the data size by filtering uninteresting data (again, halving the data size will reduce time usage by an eightfold), or we must develop and implement more efficient algorithms. This is often done by using heuristics which reduces time and space complexity by only promising an approximation of the correct answer.

Developing software for microarray analysis has proven to be a challenge much because of the many different microarray vendors and their constant modifications and development of new standards. We have overcome some problems by keeping our software up to date through an automatic software update system. This automatically adds new functionality and software patches for users connected to the internet. The users' choice of technology and methods are often changing, meaning that a software package will soon be outdated if not flexible enough to adapt the new microarray "fashions" quickly.

Developing novel methods for microarray data analysis is still an important task in bioinformatics. The reason why this field has not yet converged with accepted standard methods is the constant improvement of technology and addition of new knowledge about genes and genomes. For instance, new unsupervised clustering methods have been frequently proposed in various microarray and bioinformatics journals for many years. With the large amount of metabolic pathway and gene ontology information available in various databases, clustering methods are now used together with this information to perform a so-called semi-supervised clustering. This seems to be a trend in new algorithms and a challenge for developers. Collecting information from various databases around the world and keeping it updated and formatted before supplying it to analysis methods is difficult but essential. Future work in algorithm development for microarray analysis and decision making (e.g. deciding genes relevant to phenotypic changes) should include as much relevant information as possible.

Analysis methods where data from multiple experiments is analyzed together to reveal cross-experimental relations and global patterns between genes and samples is left for future work. Publications where this has been done do exist (e.g. Rhodes et. al. 2004 [80]), but there is still a need for this information to be efficiently used in the data analysis software.

The J-Express software has much room for improvement. More downstream data analysis methods such as Support Vector Machines (SVM), and Partial Least Squares (PLS) should be added and new options for data transformation (variance stabilization) and normalization is needed. We are also working on a framework for cross validation and classification.

8 Discussion

Microarray technology has come a long way since we started working on analysis methods in 1998. Reproducibility has increased, collaborators have moved between microarray vendors several times and heaps of analysis methods have been proposed. When looking back, we find applied methods in our projects that can be replaced by better ones in almost every phase of the analysis, even for projects only a couple of years old. For instance, in the Sudan-Norway carcinoma analysis [81] we were uncertain about how to handle the large variation between samples when looking for genes differentially expressed between tumor and normal tissue. In the beginning, we included a negative variance factor in the equation, but we soon found that this variation was really not important as long as the expression values were significantly over- or under expressed between tumor and normal samples. Today, we find many methods for microarray data analysis perhaps better suited for this analysis, such as regularized t-scores and SAM.

In the papers included in this thesis, we present novel methods for preparing and analyzing microarray gene expression data. In addition, we present projects where analysis methods have been applied to new datasets provided by collaborators. We have shown that even though the underlying data is noisy, there are still ways to bring out genes and show that they have a statistical significance. When the underlying data is of high quality, the analysis methods are simpler and easier to validate. It is tempting to suggest that whenever the data is of high quality, fewer samples are needed to discover significant biological meaning. This does however not mean that as long as there are enough data of low quality, we can always bring out the "gold" from the dross.

A recurring issue when dealing with microarray data is how to prepare the data for downstream expression analysis. The importance of preprocessing are tightly linked to the accuracy of the microarray technology. As long as the measurements are less than perfect, preprocessing will be a crucial step in the analysis. Another important factor in the preparation phase is how to get the most out of the image analysis. Most computer software packages for image processing and analysis provide a wide collection of statistics. This information should be carefully processed to optimize result quality. Studies show that different habits in filtering, imputation and normalization can give fundamentally different results.

Filtering in the data preparation step usually includes removing bad spots. The definition of a bad spot is however not always agreed upon. We have shown that patterns in the dataset can be used to impute filtered spots and further work in this field should explore what kind of filtering is best suited for imputation. For instance, spots damaged by dust and scratches and thereby filtered and imputed may result in better results than spots filtered due to weak signals (low expression).

When calculating a success rate for the various imputation methods, we hid values for imputation arbitrarily in the expression matrix and compared an imputed value to the hidden value. The rate of missing values is however not necessarily randomly distributed across genes. A better comparison could be to test the success rate only for genes that are more likely to contain missing values, such as genes with low intensities across samples. In addition, using external information such as functional groups could further increase similarity confidence when calculating correlation between genes to use in correlation based imputation.

The vast amount of methods available for expression analysis illustrates the microarray range of applications and complexity. It also seems that the imperfect nature of a microarray experiment appeal to a variety of academic fields, including statisticians, mathematicians and computer scientists. We have implemented many methods for microarray data analysis and found strengths and weaknesses in many of them. Generally, it seems that most methods produce good results when applied to high quality data. In addition, strong signals such as a high average, low within group variance fold change between two groups are always picked up in a discriminant study, while other signals such as a similar fold change but high within group variance need special treatments (for instance regularized t-score vs. traditional t-score).

For any discriminant study, a permutation test should be applied to establish a background score. Again, the inaccurate nature of the expression matrix demands results to be verified statistically.

Another recurring issue in microarray experiments is the biologist’s failure to bring in the statistical and bioinformatics expertise at an early enough stage. An example is the Sudan-Norway carcinoma analysis [81] where the experimental design and most of the hybridizations were already done at the time our group was involved. In addition, samples were harvested after the initial hybridizations were processed and microarrays had to be bought from different batches as the experiment grew in size. Different parts of the experiment were hybridized using different designs (common reference and direct hybridizations) which made direct comparison between all samples impossible. Getting results from this dataset proved to be a tremendous challenge, but the applied analysis did result in some significant genes and biological processes. For optimal results, statisticians should be included as early as possible in the idea and planning phase of a microarray experiment. This can also reduce the cost of materials and work hours.

BIBLIOGRAPHY

1. Lodish H: **Molecular cell biology**, 5th edn. New York: Freeman; 2003.
2. Nelson DL, Cox MM, Lehninger AL: **Lehninger principles of biochemistry**, 3rd edn. New York: Worth Publishers; 2000.
3. Jeong H, Tombor B, Albert R, Oltvai ZN, Barabasi AL: **The large-scale organization of metabolic networks**. *Nature* 2000, **407**(6804):651-654.
4. Jeong H, Mason SP, Barabasi AL, Oltvai ZN: **Lethality and centrality in protein networks**. *Nature* 2001, **411**(6833):41-42.
5. Kitano H: **Foundations of systems biology**. Cambridge, Mass.: MIT Press; 2001.
6. Pilpel Y, Sudarsanam P, Church GM: **Identifying regulatory networks by combinatorial analysis of promoter elements**. *Nat Genet* 2001, **29**(2):153-159.
7. Bolouri H, Bower JM: **Computational modeling of genetic and biochemical networks**. Cambridge, Mass.: MIT Press; 2001.
8. Slonim DK: **From patterns to pathways: gene expression data analysis comes of age**. *Nat Genet* 2002, **32 Suppl**:502-508.
9. Krieger CJ, Zhang P, Mueller LA, Wang A, Paley S, Arnaud M, Pick J, Rhee SY, Karp PD: **MetaCyc: a multiorganism database of metabolic pathways and enzymes**. *Nucleic Acids Res* 2004, **32**(Database issue):D438-442.
10. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT *et al*: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium**. *Nat Genet* 2000, **25**(1):25-29.
11. Camon E, Magrane M, Barrell D, Binns D, Fleischmann W, Kersey P, Mulder N, Oinn T, Maslen J, Cox A *et al*: **The Gene Ontology Annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro**. *Genome Res* 2003, **13**(4):662-672.
12. Harris MA, Clark J, Ireland A, Lomax J, Ashburner M, Foulger R, Eilbeck K, Lewis S, Marshall B, Mungall C *et al*: **The Gene Ontology (GO) database and informatics resource**. *Nucleic Acids Res* 2004, **32**(Database issue):D258-261.
13. Liebler DC: **Introduction to proteomics : tools for the new biology**. Totowa, N.J.: Humana Press; 2002.
14. Trayhurn P: **Northern blotting**. *Proc Nutr Soc* 1996, **55**(1B):583-589.
15. Southern EM: **Detection of specific sequences among DNA fragments separated by gel electrophoresis**. *J Mol Biol* 1975, **98**(3):503-517.
16. Irving RA, Hudson PJ: **Proteins emerge from disarray**. *Nat Biotechnol* 2000, **18**(9):932-933.
17. de Wildt RM, Mundy CR, Gorick BD, Tomlinson IM: **Antibody arrays for high-throughput screening of antibody-antigen interactions**. *Nat Biotechnol* 2000, **18**(9):989-994.
18. Cahill DJ: **Protein and antibody arrays and their medical applications**. *J Immunol Methods* 2001, **250**(1-2):81-91.
19. Gibson UE, Heid CA, Williams PM: **A novel method for real time quantitative RT-PCR**. *Genome Res* 1996, **6**(10):995-1001.
20. Brown M, Grundy W, Lin D, Christianini N, Sugnet C, Ares M, Haussler D: **Support vector machine classification of microarray gene expression data.**

Technical Report UCSC-CRL 99-09 University of California, Santa Cruz CA 1999.
21. DeRisi JL, Iyer VR, Brown PO: **Exploring the metabolic and genetic control of gene expression on a genomic scale**. *Science* 1997, **278**(5338):680-686.
22. Lockhart DJ, Dong H, Byrne MC, Follettie MT, Gallo MV, Chee MS, Mittmann M, Wang C, Kobayashi M, Horton H *et al*: **Expression monitoring by hybridization to high-density oligonucleotide arrays**. *Nat Biotechnol* 1996, **14**(13):1675-1680.
23. Schena M, Shalon D, Davis RW, Brown PO: **Quantitative monitoring of gene expression patterns with a complementary DNA microarray**. *Science* 1995, **270**(5235):467-470.
24. Front Line Strategic Consulting I: **DNA Microarrays, A Strategic Market Analysis. Report #1160**. 2001.
25. Front Line Strategic Consulting I: **Protein and Tissue/Cellular Arrays. Opportunities and Technical Analysis. Report #1400**. 2003.
26. Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X *et al*: **Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling**. *Nature* 2000, **403**(6769):503-511.
27. Kohane IS, Kho AT, Butte AJ: **Microarrays for an integrative genomics**. Cambridge, Mass.: MIT Press; 2003.
28. Perou CM, Sorlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Akslen LA *et al*: **Molecular portraits of human breast tumours**. *Nature* 2000, **406**(6797):747-752.
29. Bittner M, Meltzer P, Chen Y, Jiang Y, Seftor E, Hendrix M, Radmacher M, Simon R, Yakhini Z, Ben-Dor A *et al*: **Molecular classification of cutaneous malignant melanoma by gene expression profiling**. *Nature* 2000, **406**(6795):536-540.
30. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA *et al*: **Molecular classification of cancer: class discovery and class prediction by gene expression monitoring**. *Science* 1999, **286**(5439):531-537.
31. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ: **Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays**. *Proc Natl Acad Sci U S A* 1999, **96**(12):6745-6750.
32. Singh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, Tamayo P, Renshaw AA, D'Amico AV, Richie JP *et al*: **Gene expression correlates of clinical prostate cancer behavior**. *Cancer Cell* 2002, **1**(2):203-209.
33. Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C *et al*: **Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks**. *Nat Med* 2001, **7**(6):673-679.
34. Pomeroy SL, Tamayo P, Gaasenbeek M, Sturla LM, Angelo M, McLaughlin ME, Kim JY, Goumnerova LC, Black PM, Lau C *et al*: **Prediction of central nervous system embryonal tumour outcome based on gene expression**. *Nature* 2002, **415**(6870):436-442.

35. Cutler DJ, Zwick ME, Carrasquillo MM, Yohn CT, Tobin KP, Kashuk C, Mathews DJ, Shah NA, Eichler EE, Warrington JA *et al*: **High-throughput variation detection and genotyping using microarrays**. *Genome Res* 2001, **11**(11):1913-1925.
36. Solinas-Toldo S, Lampel S, Stilgenbauer S, Nickolenko J, Benner A, Dohner H, Cremer T, Lichter P: **Matrix-based comparative genomic hybridization: biochips to screen for genomic imbalances**. *Genes Chromosomes Cancer* 1997, **20**(4):399-407.
37. Pinkel D, Segraves R, Sudar D, Clark S, Poole I, Kowbel D, Collins C, Kuo WL, Chen C, Zhai Y *et al*: **High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays**. *Nat Genet* 1998, **20**(2):207-211.
38. White KP, Rifkin SA, Hurban P, Hogness DS: **Microarray analysis of Drosophila development during metamorphosis**. *Science* 1999, **286**(5447):2179-2184.
39. Bozdech Z, Llinas M, Pulliam BL, Wong ED, Zhu J, DeRisi JL: **The transcriptome of the intraerythrocytic developmental cycle of Plasmodium falciparum**. *PLoS Biol* 2003, **1**(1):E5.
40. Simon RM, Dobbin K: **Experimental design of DNA microarray experiments**. *Biotechniques* 2003, **Suppl**:16-21.
41. Simon R: **Design and analysis of DNA microarray investigations**. New York: Springer; 2003.
42. Dobbin K, Simon R: **Comparison of microarray designs for class comparison and class discovery**. *Bioinformatics* 2002, **18**(11):1438-1445.
43. Dobbin K, Shih JH, Simon R: **Questions and answers on design of dual-label microarrays for identifying differentially expressed genes**. *J Natl Cancer Inst* 2003, **95**(18):1362-1369.
44. Glonek GF, Solomon PJ: **Factorial and time course designs for cDNA microarray experiments**. *Biostatistics* 2004, **5**(1):89-111.
45. Wit E, McClure J: **Statistics for microarrays : design, analysis and inference**. Chichester: Wiley; 2004.
46. Kerr MK, Churchill GA: **Experimental design for gene expression microarrays**. *Biostatistics* 2001, **2**(2):183-201.
47. Churchill GA: **Fundamentals of experimental design for cDNA microarrays**. *Nat Genet* 2002, **32 Suppl**:490-495.
48. Townsend JP: **Multifactorial experimental design and the transitivity of ratios with spotted DNA microarrays**. *BMC Genomics* 2003, **4**(1):41.
49. Yang YH, Buckley MJ, Dudoit S, Speed TP: **Comparison of methods for image analysis on cDNA microarray data**. *Journal of Computational and Graphical Statistics* 2002, **11**:108-136.
50. Bolstad BM, Irizarry RA, Astrand M, Speed TP: **A comparison of normalization methods for high density oligonucleotide array data based on variance and bias**. *Bioinformatics* 2003, **19**(2):185-193.
51. Yang YH, Dudoit S, Luu P, Lin DM, Peng V, Ngai J, Speed TP: **Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation**. *Nucleic Acids Res* 2002, **30**(4):e15.

52. Quackenbush J: **Microarray data normalization and transformation**. *Nat Genet* 2002, **32 Suppl**:496-501.
53. van de Peppel J, Kemmeren P, van Bakel H, Radonjic M, van Leenen D, Holstege FC: **Monitoring global messenger RNA changes in externally controlled microarray experiments**. *EMBO Rep* 2003, **4**(4):387-393.
54. Workman C, Jensen LJ, Jarmer H, Berka R, Gautier L, Nielser HB, Saxild HH, Nielsen C, Brunak S, Knudsen S: **A new non-linear normalization method for reducing variability in DNA microarray experiments**. *Genome Biol* 2002, **3**(9):research0048.
55. Smyth GK, Speed T: **Normalization of cDNA microarray data**. *Methods* 2003, **31**(4):265-273.
56. Durbin BP, Hardin JS, Hawkins DM, Rocke DM: **A variance-stabilizing transformation for gene-expression microarray data**. *Bioinformatics* 2002, **18 Suppl 1**:S105-110.
57. Huber W, von Heydebreck A, Sultmann H, Poustka A, Vingron M: **Variance stabilization applied to microarray data calibration and to the quantification of differential expression**. *Bioinformatics* 2002, **18 Suppl 1**:S96-104.
58. Geller SC, Gregg JP, Hagerman P, Rocke DM: **Transformation and normalization of oligonucleotide microarray data**. *Bioinformatics* 2003, **19**(14):1817-1823.
59. Durbin BP, Rocke DM: **Variance-stabilizing transformations for two-color microarrays**. *Bioinformatics* 2004, **20**(5):660-667.
60. Jornsten R, Wang HY, Welsh WJ, Ouyang M: **DNA microarray data imputation and significance analysis of differential expression**. *Bioinformatics* 2005, **21**(22):4155-4161.
61. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB: **Missing value estimation methods for DNA microarrays**. *Bioinformatics* 2001, **17**(6):520-525.
62. Bo TH, Dysvik B, Jonassen I: **LSimpute: accurate estimation of missing values in microarray data with least squares methods**. *Nucleic Acids Res* 2004, **32**(3):e34.
63. Raychaudhuri S, Sutphin PD, Chang JT, Altman RB: **Basic microarray analysis: grouping and feature reduction**. *Trends Biotechnol* 2001, **19**(5):189-193.
64. Eisen MB, Spellman PT, Brown PO, Botstein D: **Cluster analysis and display of genome-wide expression patterns**. *Proc Natl Acad Sci U S A* 1998, **95**(25):14863-14868.
65. Tusher VG, Tibshirani R, Chu G: **Significance analysis of microarrays applied to the ionizing radiation response**. *Proc Natl Acad Sci U S A* 2001, **98**(9):5116-5121.
66. Blum A, Langley P: **Selection of relevant features and examples in machine learning**. *Artificial Intelligence* 1997:245-271.
67. Model F, Adorjan P, Olek A, Piepenbrock C: **Feature selection for DNA methylation based cancer classification**. *Bioinformatics* 2001, **17 Suppl 1**:S157-164.

68. Li T, Zhang C, Ogihara M: **A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression**. *Bioinformatics* 2004, **20**(15):2429-2437.
69. Dudoit S, Fridlyand J, Speed T: **Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data**. *Technical report 576, Mathematical Sciences Research Institute, Berkeley, CA* 2000.
70. Schölkopf B, Smola AJ: **Learning with kernels : support vector machines, regularization, optimization, and beyond**. Cambridge, Mass.: MIT Press; 2002.
71. Brown MP, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Ares M, Jr., Haussler D: **Knowledge-based analysis of microarray gene expression data by using support vector machines**. *Proc Natl Acad Sci U S A* 2000, **97**(1):262-267.
72. Mao Y, Zhou X, Pi D, Sun Y, Wong ST: **Multiclass cancer classification by using fuzzy support vector machine and binary decision tree with gene selection**. *J Biomed Biotechnol* 2005, **2005**(2):160-171.
73. Grant GR, Liu J, Stoeckert CJ, Jr.: **A practical false discovery rate approach to identifying patterns of differential expression in microarray data**. *Bioinformatics* 2005, **21**(11):2684-2690.
74. Benjamini Y, Hochberg Y: **Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing** *Journal of the Royal Statistical Society Series B (Methodological)* 1995, **57**(1):289-300.
75. Pavlidis P, Lewis D, Noble W: **Exploring gene expression data with class scores**. *Pacific Symposium on Biocomputing* 2002:474-485.
76. Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, Stoeckert C, Aach J, Ansorge W, Ball CA, Causton HC *et al*: **Minimum information about a microarray experiment (MIAME)-toward standards for microarray data**. *Nat Genet* 2001, **29**(4):365-371.
77. Spellman PT, Miller M, Stewart J, Troup C, Sarkans U, Chervitz S, Bernhart D, Sherlock G, Ball C, Lepage M *et al*: **Design and implementation of microarray gene expression markup language (MAGE-ML)**. *Genome Biol* 2002, **3**(9).
78. Dysvik B, Petersen K, Jonassen I: **Integrating MAGE in J–Express using Java MAGEstk. Poster presented at MGED8, 11-13 september 2005**. 2005.
79. Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B: **Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization**. *Mol Biol Cell* 1998, **9**(12):3273-3297.
80. Rhodes DR, Yu J, Shanker K, Deshpande N, Varambally R, Ghosh D, Barrette T, Pandey A, Chinnaiyan AM: **Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression**. *PNAS* 2004, **101**(25):9309-9314.
81. Dysvik B, Vasstrand EN, Lovlie R, Elgindi OA, Kross KW, Aarstad HJ, Johannessen AC, Jonassen I, Ibrahim SO: **Gene expression profiles of head and neck carcinomas from Sudanese and Norwegian patients reveal common biological pathways regardless of race and lifestyle**. *Clin Cancer Res* 2006, **12**(4):1109-1120.