Multivariate Analysis of Gene Expression Data and Functional Information: Automated Methods for Functional Genomics

Andreas Rechtsteiner

29th April 2005

Contents

Chapter 1

Introduction

1.1 Overview and Motivation of Work

This dissertation is comprised of several manuscripts[1] of my PhD work on developing new algorithms for gene expression analysis and automated mining of functional information from literature for Bioinformatics. Here in the introduction I will give a high level overview of the work, present its motivation, some of the necessary technical background and how the different parts of the work relate to each other.

1.1.1 Gene Expression Analysis Work

Biology used to be a mainly hypothesis driven science in which experiments were carefully designed to answer one or few very specific questions, like the function of a specific protein in a specific context. The development and success of molecular biology and genetics, combined with computer technology, has lead to the emergence of techniques that draw on inferences from large amounts of data derived from so-called *high-throughput experiments* [Lander, 1996, Lander, 1999, Zweiger, 1999, D'haeseleer, 2000]. These experiments allow for the analysis and monitoring of many cellular components, e.g. genes and proteins, in parallel. The genome sequencing projects marked the first step into this new, data-rich and inference driven era, i.e. the new high-throughput sequencing technologies allowed the focus of sequencing to shift from individual genes of interest to the whole genome of an organism. New high-throughput technologies, many of which are based on the sequencing results, are being developed. These technologies enable the monitoring of many genes or proteins and their interactions in parallel to obtain a system level perspective

[1] Some already published or in part presented at conferences, some in preparation for publication [Wall et al., 2003, Challacombe et al., 2004, Rechtsteiner et al., 2003, Rechtsteiner and Rocha, 2004a, Rechtsteiner and Rocha, 2004b, Rechtsteiner et al., 2005].

of cellular processes. The terms *Functional Genomics* and *Systems Biology* describe research attempting such global characterization and understanding of cellular behavior [Kitano, 2001, Ideker et al., 2001, Kanehisa, 2000]. The development of algorithms to mine the large amounts of data from high-throughput experiments for biological information is the main area of research within the field of *Bioinformatics*.

The first part of this dissertation is concerned with algorithms for analysis of data from *microarray* experiments to measure mRNA or gene transcript levels [Fodor et al., 1993, Schena et al., 1995] (see appendix A.1 for some background on the biology of gene expression). Microarrays allow for the simultaneous measurement of the expression levels of tens of thousands of genes, sometimes the whole genome, of an organism. One of the fundamental ways in which a cell regulates its biological processes and responds to changing conditions is by regulating the expression of its genes. Large scale gene expression measurements can be used for determining the functions of newly identified genes, for obtaining *genetic fingerprints* for diseases and for getting a broader, system-level understanding of life on the cellular level [Lander, 1999, Ideker et al., 2001]. Two time series data sets that will be discussed specifically in this dissertation are gene expression data from human fibroblast cells that were infected with a herpes virus [Browne et al., 2001, Challacombe et al., 2004] and expression data obtained during the cell-cycle of yeast [Cho et al., 1998][2]. For the herpes data, the goal of the study is to identify the genes whose expression responds significantly to the virus infection, what their expression response patterns are and what their functions in the context of the virus infection are. For the yeast cell-cycle study the goal is to identify the genes that are significantly cell-cycle regulated, i.e. identify genes that have periodic expression patterns in synchrony with the cell-cycle, and identify which genes are induced (increased expression) in which phase of the cell-cycle.

To answer such questions, much of the multi-variate data analysis of gene expression data has focused on clustering techniques. Examples are hierarchical clustering [Eisen et al., 1998], K-means [Tavazoie et al., 1999], Self-Organizing Maps (SOM) [Tamayo et al., 1999] and the fitting of Multi-Variate Gaussian Mixture Models (Mclust algorithm) [Yeung et al., 2001a] (see appendix B.4 for brief discussions of these algorithms and some of their applications). The assumption underlying the use of clustering techniques is that a group of genes participating in the same biological process is similarly expressed because their proteins will be in similar demand. In chapter 4 concerns with the application of clustering algorithms to time series gene expression data will be discussed. It has been shown that clusters obtained with different clustering algorithms for time series expression data are not as distinct and discrete in expression space as one might expect and might be desirable when applying clustering algorithms. A problem for some clustering algorithms

[2] The herpes data is analyzed in more detail, the cell-cycle data is used to illustrate the developed clustering algorithm.

is also that they require the number of clusters in the data to be known. Examples of algorithms needing such parameters are K-means, SOM and the Mclust algorithm. Both problems might be helped with proper visualization of the expression data. Here we present Singular Value Decomposition (SVD) [Press et al., 1992, Golub and Van Loan, 1996] as an algorithm that can help in such visualization. SVD is discussed in detail in chapter 2 and examples of different applications of SVD are given. Chapter 3 presents the analysis of a gene expression data set of herpes virus infected human fibroblast cells [Browne et al., 2001] with SVD. The study identified two clusters of genes with very different response patterns. The biological significance of the genes in the two clusters was assessed manually, by a biologist. This assessment validated the derived clusters.

The findings of this study and insights we gained from others [Holter et al., 2000, Alter et al., 2000, Raychaudhuri et al., 2000, Cho et al., 1998, Tavazoie et al., 1999, Tamayo et al., 1999] motivated work on two new algorithms for gene expression analysis based on SVD and the subspaces in expression space it can identify. The motivations and the algorithms are presented in chapter 4. The first algorithm uses the distribution of the polar angles of genes projected into two-dimensional SVD subspaces to identify genes that are significantly expressed. The second algorithm clusters these significantly expressed genes based on the density of the distribution of their polar angles. The application of the algorithms to a well studied expression data set is presented, the yeast cell-cycle data set of [Cho et al., 1998]. Chapter 5 presents the application of the algorithms to the herpes data set from chapter 3.

1.1.2 Automated Information Retrieval from Literature for Computational Biology

In chapter 5 and 6 work on automated mining of biological information from literature is presented. This work was motivated by the gene expression analysis work presented in the earlier chapters. Although gene expression analysis provides useful insights to biologists, the biological meaning of numerical gene expression analysis results is often not obvious. Co-expression clusters can contain hundreds of genes, as do the clusters presented in chapter 4 for the yeast cell-cycle data and in chapter 5 for the herpes virus infected human fibroblast cells. Although databases like GenBank [Benson et al., 2004, NCBI, 2004] and SwissProt/UniProt [Bairoch et al., 2005, SIB/EBI, 2004] exist which contain functional annotations for individual genes and proteins, it is difficult to identify the significant biological function, what we frequently term a *functional theme*, for large numbers of genes in the context of an experiment. Further, finding such functional themes from individual protein or gene annotations manually, often requires expert knowledge. Whereas much of the annotations in databases is free text, progress is being made in standardizing annotations, for example by the Gene Ontology (GO) consortium [The Gene Ontology Consortium, 2004, Harris et al., 2004] which is developing a hierarchical ontology for annotation of genes and proteins. But as the annotation of the clusters in chapter 3 with GO by a biological expert showed, identifying functional themes for large groups of genes and proteins still proves difficult.

In chapter 5 a algorithm is presented that will assist in identifying functional themes for groups of genes or proteins (e.g. genes from co-expression clusters) in an automated fashion. The *vector space model* from *Information Retrieval* (IR) [Baeza-Yates et al., 1999] was adapted to represent and mine knowledge in the bio-medical literature database MEDLINE for information about clusters of genes. The vector space model is used in IR for indexing and retrieval of documents based on keyword queries. Here the Medical Subject Headings (MeSH) are used as the keyword vocabulary. The presented technique identifies functional themes that are associated with groups of genes in the literature. An application to the gene expression clusters of the herpes data is presented.

Although our results in chapter 5 show the validity of the developed method, we present a more quantitative validation in chapter 6. Here a large-scale study on how well the developed method can classify proteins into known protein sequence families (the Pfam family classification) is presented. Over 15,000 proteins are classified into 1600 different families based on 26,000 publications.

In section 1.2 a short description of the different array technologies used to measure gene expression is presented. In appendix B an overview of data analysis techniques typically applied to gene expression data is given.

1.2 Microarrays for Transcript Level Measurements on a Genomic Scale[3]

Microarrays are extensions of *hybridization* (see Glossary) based methods like Southern and Northern blots that have been used for decades to identify and quantify individual nucleic acid sequences in biological samples [Knudsen, 2002]. Hybridization is the process by which two complementary nucleic acid sequences, like DNA or RNA strands, interact so that double-stranded structures are formed. Complementary sequences are nucleic acid sequences that can form such double-stranded structures with each other by following base-pairing rules (in DNA adenine (A) pairs with thymine (T) and cytosine (C) with guanine (G) so that the complementary sequence of GTAC would be CATG). The main novelty of microarray technology is the ability to measure the abundance of transcripts of thousands of genes in a single experiment with a single chip. Several developments in biology have made it possible to perform these measurements in such a highly parallel fashion.

[3] The terms *transcript levels*, *expression levels* and *mRNA levels* will be used interchangeably. Similarly will the terms *microarray* and *chip*, or *DNA chip*, be used interchangably.

Large-scale genome sequencing projects have made it possible to assemble collections of DNAs that correspond to all, or a large fraction of, the genes in many organisms from bacteria to humans. Second, technical advances have made it possible to generate arrays with very high densities of DNA probes, allowing for tens of thousands of genes to be represented on standard glass microscope slides or similar sized chips. Finally, advances in fluorescent labeling of nucleic acids and fluorescent detection have made the use of these arrays simpler and more accurate.

Two main microarray technologies have emerged, oligonucleotide arrays [Fodor et al., 1993] and cDNA microarrays [Schena et al., 1995, Duggan et al., 1999, DeRisi et al., 2000]. All microarrays have DNA nucleotide strands from genes to be assayed, called the *probes,* fixated at known positions on the chip. The preparation of the pool of mRNA nucleotide strands to be assayed, the *target* strands, as well as the process of measuring the abundance of individual mRNAs, are similar for all microarrays as well. The target mRNAs are reverse transcribed to cDNA and are fluorescently labeled in the process. After hybridization to the probe strands, the abundance of the different mRNAs is measured by the intensity of the fluorescent signal at the known probe strand locations. Some of the details and differences between cDNA and oligonucleotide microarrays are discussed in the next two sections.

1.2.1 cDNA Microarrays

One of the differences between cDNA microarrays and oligonucleotide arrays is that for the former the fixated probes are cDNA strands. cDNA stands for *complementary DNA*, a single stranded DNA molecule that is complementary to a full-length mRNA, typically 500-5000 bases long. cDNA probes are placed on a coated glass microscope slide using a computer-controlled robot. Besides the difference in the probes, another main difference to oligonucleotide array technology is that for cDNA microarrays, the cDNA target pool is a mixture of differently labeled cDNA from samples to be tested and from some control or reference sample (see also Fig. 1.1). The mRNA from both the test and reference sample have been reverse transcribed to cDNA and in the process fluorescently labeled with two different dyes (usually Cy-3 and Cy-5-dUTP). Both fluorescent cDNA samples are mixed and allowed to hybridize to the cDNA probes on the array. *Comparative hybridization*[4] of the test and reference cDNA target samples is supposed to take care of differences among the probe spots, like varying density of probe strands, which could bias intensity measurements and introduce so-called *spot effects*. The test and reference mRNA target samples should be affected equally by such spot effects and the ratio of fluorescent signal intensity of the test to reference sample should then be free of spot effect biases.

[4] Sometimes also referred to as *competitive hybridization*, because test sample cDNA and reference sample cDNA 'compete' for hybridization to the probes.

To measure mRNA abundance of the test and reference sample, the hybridized, fluorescent targets are excited with a laser and the spectra for the two dyes are measured using a scanner. Usually monochrome images from the scanner are imported into software in which the images are pseudo-colored and merged (Fig. 1.1 #5 and #6). The data from a hybridization experiment is then reported as an intensity ratio (Cy-3/Cy-5) in which significant deviations from 1 (no change) are indicative of increased (>1) or decreased (<1) levels of gene expression relative to the reference sample.

An advantage of cDNA microarrays over oligonucleotide arrays is that the technology is non-proprietary and less expensive. An online guide on how to build and arrayer from scratch can be found at the Brown lab at Stanford University [DeRisi et al., 2000]. More details on microarray technology can be found in [DeRisi Lab, 2005, Leming, 2002, Schena et al., 1995, Duggan et al., 1999, Eisen and Brown, 1999].

1.2.2 Oligonucleotide arrays

The most used oligonucleotide arrays are commercially produced and distributed by Affymetrix [Affymetrix, 2005, Fodor et al., 1993]. The discussion here focuses on the technology of these often called *Affymetrix* or just *Affy chips*. Oligonucleotide arrays [Fodor et al., 1993] use *oligonucleotides*, relatively short sequences of 20 to 25 nucleotides, as probe strands on the chips. In Affy chips these probes are synthesized directly onto the chip surface using a combination of semiconductor-based photolithography and light-directed chemical synthesis[5]. The main difference of the Affy chip technology to cDNA microarray technology is that a gene's expression level is not measured with one kind of nucleotide probe strand, e.g. one type of cDNA probe strand at each spot as for cDNA microarrays. A set of typically about 20 different oligonucleotide probe pairs of length 20 to 25 are used to assay the expression level of a single gene. Affymetrix therefore introduced a slightly different terminology than has been used for cDNA microarrays. The 20 different oligonucleotide pairs used to assay a gene's expression level are referred to as *probes* or *probe pairs*, the 20 different probe pairs for a specific gene are referred to as a *probe set*. One of the oligonucleotides of each probe pair is a *perfect match* to a gene's sequence, the other is a so-called *mismatch* sequence where the center nucleotide has been altered. The mismatch sequence serves as a control and is supposed to detect background or noise signal due to non-specific hybridization of sequences that are not from the gene to be assayed but may be similar in sequence. Affymetrix's analysis software GeneChip calculates an expression level for each probe pair by subtracting the mismatch value from the perfect match value. An expression value for each gene, or probe set,

[5] Other technologies for producing oligonucleotide chips, for example based on ink-jet printer technology, have been developed [Stekel, 2003]

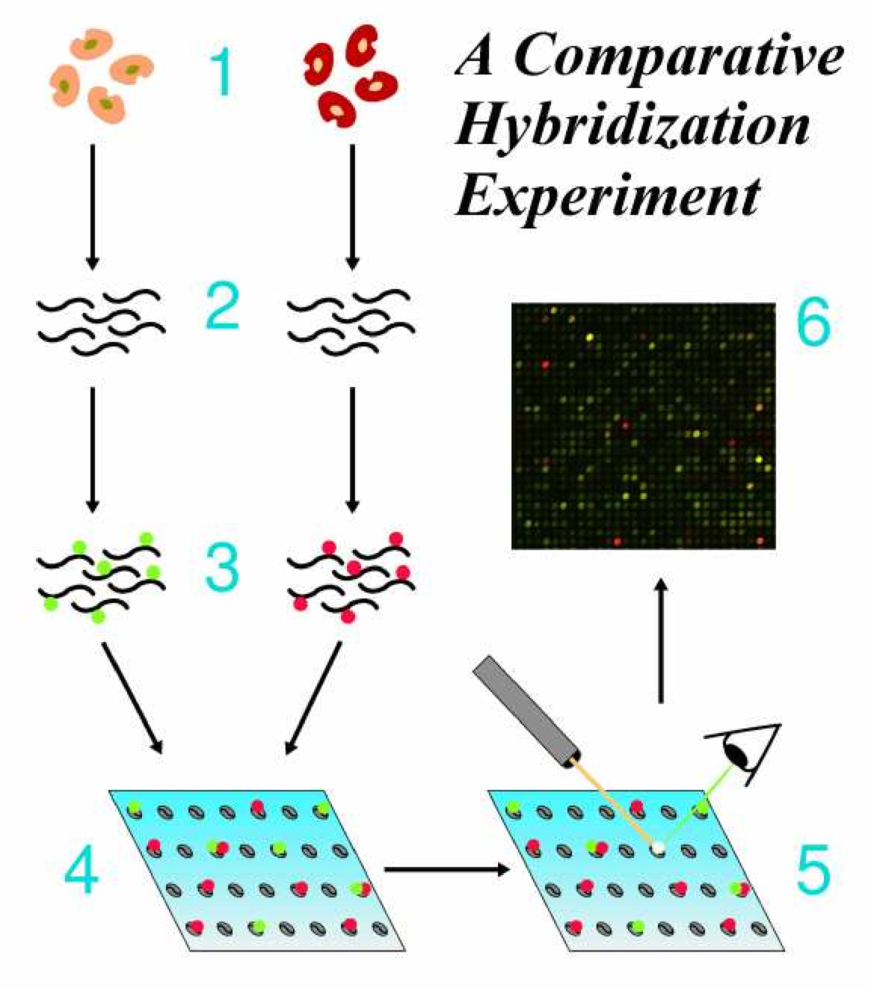


Figure 1.1: A typical cDNA microarray experiment: 1.) obtain test and control cell populations. 2.) mRNA extraction. 3.) reverse transcription to cDNA and fluorescent labeling. 4.) hybridization of both samples to cDNA microarray. 5.) scanning of the hybridized array. 6.) the resulting image. Figure adapted from http://www.cs.wustl.edu/~jbuhler/research/array/

is calculated by simple averaging the 20 probe pair differences with extreme value, or outlier, removal.

Chapter 2

Singular Value Decomposition for Gene Expression Analysis[1]

As outlined above, one of the challenges in current bioinformatics is to develop effective ways to analyze global gene expression data. In addition to being of a broader utility in analysis methods, Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) can be valuable tools for characterizing the structure of the data. SVD and PCA are common techniques for analysis of multivariate data, and gene expression data are well suited to analysis using SVD/PCA. A single microarray experiment can generate measurements for thousands of genes. Present experiments typically consist of around a dozen assays, but can consist of hundreds [Hughes et al., 2000]. Gene expression data are currently rather noisy, and SVD can detect and extract small signals from noisy data. The goal of this chapter is to provide precise explanations of the use of SVD and PCA for gene expression analysis, illustrating methods using simple examples. Our aims are 1) to provide specific examples of the application of SVD methods and interpretation of their results; 2) to establish a foundation for understanding previous applications of SVD to gene expression analysis; and 3) to provide interpretations and references to related work that may inspire new advances.

In section 2.1, the SVD is defined, with comparisons to other methods described. A summary of previous applications is presented in order to suggest directions for SVD analysis of gene expression data. In section 2.3 we discuss applications of SVD to gene expression analysis, including specific methods for SVD-based visualization of gene expression data, and use of SVD in detection of weak expression patterns. Some examples are given of previous applications of SVD to analysis of gene expression data. The discussion in section 2.4 gives some general advice on the use of SVD analysis on gene expression data, and includes references to specific published SVD-based methods for gene expression analysis. Finally, in section 2.5 we provide information on some

[1] This chapter has been adapted from [Wall et al., 2003].

available resources and further reading.

2.1 Mathematical definition of the SVD

Let X denote an $m \times n$ matrix of real-valued data with rank r, where without loss of generality $m \geq n$, and therefore $r \leq n$. In the case of microarray data, x_{ij} is the expression level of the ith gene in the jth assay. The elements of the ith row of X form the n-dimensional vector $\mathbf{g}_i$, which we refer to as the *expression vector* or *transcriptional response* of the ith gene. Alternatively, the elements of the jth column of X form the m-dimensional vector $\mathbf{a}_j$, which we refer to as the *expression profile* of the jth assay. The equation for the singular value decomposition of X is the following:

$$X = USV^T \tag{2.1}$$

where U is an $m \times n$ matrix, S is an $n \times n$ diagonal matrix, and V^T is also an $n \times n$ matrix. The columns of U are called the *left singular vectors*, $\{\mathbf{u}_k\}$, and form an orthonormal basis for the assay expression profiles, so that $\mathbf{u}_i \mathbf{u}_j = 1$ for $i = j$, and $\mathbf{u}_i \mathbf{u}_j = 0$ otherwise. The rows of V^T contain the elements of the *right singular vectors*, $\{\mathbf{v}_k\}$, and form an orthonormal basis for the gene expression vectors. The elements of S are zero everywhere except on the diagonal. The elements on the diagonal are called the *singular values*. Thus, $S = diag(s_1, ..., s_n)$. Furthermore, $s_k > 0$ for $1 \leq k \leq r$, and $s_k = 0$ for $(r+1) \leq k \leq n$. By convention, the ordering of the singular vectors is determined by high-to-low sorting of singular values, with the highest singular value in the upper left index of the S matrix.

Closest Rank-l Approximation. One important result of the SVD of X is that the matrix $X^{(l)}$ in Eqn. 2.2 is the closest rank-l approximation of X in the sense that it minimizes the sum of the squares of the residuals of the matrix elements.

$$X^{(l)} = \sum_{k=1}^{l} \mathbf{u}_k s_k \mathbf{v}_k^T \tag{2.2}$$

Calculation of SVD. One way to calculate the SVD is to first calculate V^T and S by diagonalizing $X^T X$:

$$X^T X = VS^2V^T \tag{2.3}$$

and then to calculate U as follows:

$$U = XVS^{-1} \tag{2.4}$$

where the null space of X spanned by the *(r+1),...,n* columns of V is ignored in the matrix multiplication. Choices for the remaining *n-r* singular vectors in V or U (which have singular values of exactly zero) may be calculated using the Gram-Schmidt orthogonalization process or some other extension method. Note that Eqn. 2.3 also illustrates that the singular values squared correspond to the eigenvalues of matrix X^TX. In practice there are several methods for calculating the SVD that are of higher accuracy and speed. A linear algebra or matrix computation book like [Golub and Van Loan, 1996] can be consulted for such algorithms.

2.1.1 Relation to Principal Component Analysis

There is a direct relation between PCA and SVD in the case where principal components are calculated from the covariance matrix. In this case, PCA can be formulated for a matrix X with dimensions $m \times n$ and rank r the following way:

$$X = \mathbf{1}\overline{\mathbf{a}}^T + TP^T \tag{2.5}$$

where the first term is an outer vector product between **1**, a column vector of all 1's and length m, and $\overline{\mathbf{a}}^T$, a row-vector of length n containing the means of the column vectors of matrix X. T is an orthogonal $m \times n$ matrix with the column vectors being called the *principal component score vectors*, and P^T being an orthogonal $n \times n$ matrix with its row vectors being the eigenvectors of the covariance matrix of the column vectors of X:

$$V = (X - \mathbf{1}\overline{\mathbf{a}}^T)^T(X - \mathbf{1}\overline{\mathbf{a}}^T) \tag{2.6}$$

. These row vectors of P^T are called the *principal component loading vectors* or just *principal component vectors.* Typically the principal component vectors are normalized to unit length. The *eigenvalues* of the covariance matrix indicate the amount of variance captured or modeled by the corresponding eigenvectors. PCA and SVD can be directly compared if matrix X is conditioned so that its column vectors have mean zero, i.e. $\overline{\mathbf{a}} = 0$. From equations 2.1 and 2.5 it follows that

$$USV^T = TP^T \tag{2.7}$$

We know further that the row vectors of both V^T and P^T are the eigenvectors of the covariance matrix X^TX. Therefore

$$V^T = P^T$$

and it follows

$$US = T$$

Therefore, the principal component loading vectors of P^T correspond to the right-singular vectors of $V^T$and the score vectors of T correspond to the left-singular vectors of matrix U, scaled by the corresponding singular values of S. The Euclidean length of each score vector is equal to the corresponding singular value squared, equal the variance captured by the corresponding principal component vector:

$$\mathbf{t}_i\mathbf{t}_i = s_i\mathbf{u}_i s_i\mathbf{u}_i = s_i^2$$

Most PCA implementations return for a matrix X the principal component vectors and the corresponding variances (the corresponding eigenvalues of the covariance matrix). The principal component scores then have to be calculated by projection of the object vectors (rows of column centered matrix X) onto the principal component vectors:

$$T = (X - \overline{\mathbf{a}}^T)P$$

In contrast, SVD algorithms in general provide all this information without further calculations in matrices U and V^T plus the respective variances or square roots of variances in the singular value matrix S.

2.2 Illustrative Applications of SVD and PCA

SVD and PCA have found wide-ranging applications. Here we describe several that also might suggest potential applications to gene expression analysis.

Image processing and compression. The property of SVD to provide the closest rank-l approximation for a matrix X (Eqn. 2.2) is used in image processing for compression and noise reduction, a common application of SVD also in other fields. By setting the small singular values to zero, we can obtain matrix approximations whose rank equals the number of remaining singular values (see Eqn. 2.2). Each term $\mathbf{u}_k s_k \mathbf{v}_k^T$ is called a *principal image*. Very good approximations can often be obtained using only a small number of terms [Richards, 1993]. SVD is applied in similar ways to signal processing problems [Deprette, 1988].

Immunology. One way to capture global prototypical immune response patterns is to use PCA on data obtained from measuring antigen-specific IgM (dominant antibody in primary immune responses) and IgC (dominant antibody in secondary immune responses) immunoglobulins using ELISA assays. Fesel and Coutinho [Fesel and Coutinho, 1998] measured IgM and IgC responses in Lewis and Fischer rats before and at three time points after immunization with myelin basic protein (MBP) in complete Freud's adjuvant (CFA), which is known to provoke experimental allergic encephalomeyelitis (EAE). They discovered distinct and mutually independent components of IgM reaction repertoires, and identified a small number of strain-specific prototypical regulatory responses.

Molecular dynamics. PCA and SVD analysis methods have been developed for characterizing protein molecular dynamics trajectories [Garcia, 1992, Romo et al., 1995]. In a study of myoglobin, Romo et al. [Romo et al., 1995] used molecular dynamics methods to obtain atomic positions of all atoms sampled during the course of a simulation. The higher principal components of the dynamics were found to correspond to large-scale motions of the protein. Visualization of the first three principal components revealed an interesting type of trajectory that was described as resembling beads on a string, and revealed a visibly sparse sampling of the configuration space.

Small-angle scattering. SVD has been used to detect and characterize structural intermediates in biomolecular small-angle scattering experiments [Chen et al., 1996]. This study provides a good illustration of how SVD can be used to extract biologically meaningful signals from the data. Small-angle scattering data were obtained from partially unfolded solutions of lysozyme, each consisting of a different mix of folded, collapsed and unfolded states. The data for each sample was in the form of intensity values sampled at around 100 different scattering angles. UV spectroscopy was used to determine the relative amounts of folded, collapsed and unfolded lysozyme in each sample. SVD was used in combination with the spectroscopic data to extract a scattering curve for the collapsed state of the lysozyme, a structural intermediate that was not observed in isolation.

Information Retrieval. SVD became very useful in Information Retrieval (IR) to deal with linguistic ambiguity issues. IR works by producing the documents most associated with a set of keywords in a query. Keywords, however, necessarily contain much synonymy (several keywords refer to the same concept) and polysemy (the same keyword can refer to several concepts). For instance, if the query keyword is "feline", traditional IR methods will not retrieve documents using the word "cat" - a problem of synonymy. Likewise, if the query keyword is "java", documents on the topic of Java as a computer language, Java as an Island in Indonesia, and Java as a coffee bean will all be retrieved - a problem of polysemy. A technique known as Latent Semantic Indexing (LSI) [Berry et al., 1995] addresses these problems by calculating the best rank-l approximation of the keyword-document matrix using its SVD. This produces a lower dimensional space of *eigen-keywords* and *eigen-documents* (singular vectors). Each eigen-keyword can be associated with several keywords as well as particular senses of keywords. In the synonymy example above, "cat" and "feline" would therefore be strongly correlated with the same eigen-keyterm. Similarly, documents using "Java" as a computer language tend to use many of the same keywords, but not many of the keywords used by documents describing "Java" as coffee or Indonesia. Thus, in the space of singular vectors, each of these senses of "java" is associated with distinct eigen-keywords.

2.3 SVD analysis of gene expression data

In this section examples of SVD-based analysis methods as applied to gene expression data are provided. Before illustrating specific techniques, we will discuss ways of interpreting the SVD in the context of gene expression data. This interpretation and the accompanying nomenclature will serve as a foundation for understanding the methods described later. A natural question a biologist might ask is, "What is the biological significance of the SVD?" There is no general answer to this question, as it depends on the specific application. Classes of experiments can be provided as a guide for individual cases, however. Two broad classes of applications under which most studies will fall are defined: *systems biology applications*, and *diagnostic applications*. In both cases, the n columns of the gene expression data matrix X correspond to assays, and the m rows correspond to the genes. The SVD of X produces two orthonormal bases, one defined by right singular vectors and the other by left singular vectors. Referring to the definitions in section 2.1, the right singular vectors span the space of the gene expression vectors $\{\mathbf{g}_i\}$ and the left singular vectors span the space of the assay expression profiles $\{\mathbf{a}_j\}$. Following the convention of Alter et al. [Alter et al., 2000], we refer to the left singular vectors $\{\mathbf{u}_k\}$ as *eigenassays* and to the right singular vectors $\{\mathbf{v}_k\}$ as *eigengenes*. We sometimes refer to an eigengene or eigenassay generically as a singular vector, or, by analogy with PCA, as a *component*. We refer to a triplet of corresponding eigenassay, singular value and eigengene as a SVD mode. Eigengenes, eigenassays and other definitions and nomenclature in this section are depicted in Figure 2.1. In applications related to systems biology, we generally wish to understand relationships among genes. The signal of interest in this case is the gene expression vector $\mathbf{g}_i$. By Equation 2.1, a gene expression vector $\mathbf{g}_i$ can be expressed as a linear combination of the eigengenes $\{\mathbf{v}_k\}$:

$$\mathbf{g}_i = \sum_{k=1}^{r} u_{ik} s_k \mathbf{v}_k \tag{2.8}$$

The ith row of U, $\mathbf{g}_i'$ (see Fig. 2.1), contains the coordinates of the ith gene expression vector in the basis of the scaled eigengenes, $s_k\mathbf{v}_k$. Note that because V is orthonormal (as is U) it follows

$$XV = US \tag{2.9}$$

and that the new basis is a rotation of the original basis vectors.

If $r < n$, the $\mathbf{g}_i'$ are lower dimesnional than the $\mathbf{g}_i$, although they capture all the information of the $\mathbf{g}_i$. Note that due to noise in the data, $r = n$ in any real gene expression data set. Similar to other applications of SVD like in image processing or Latent Semantic Analysis (see discussions above), the last 'few' (depending on the problem and dimensionality n of the data) singular values in S are often found to be close to zero and considered as only capturing noise in the data. The

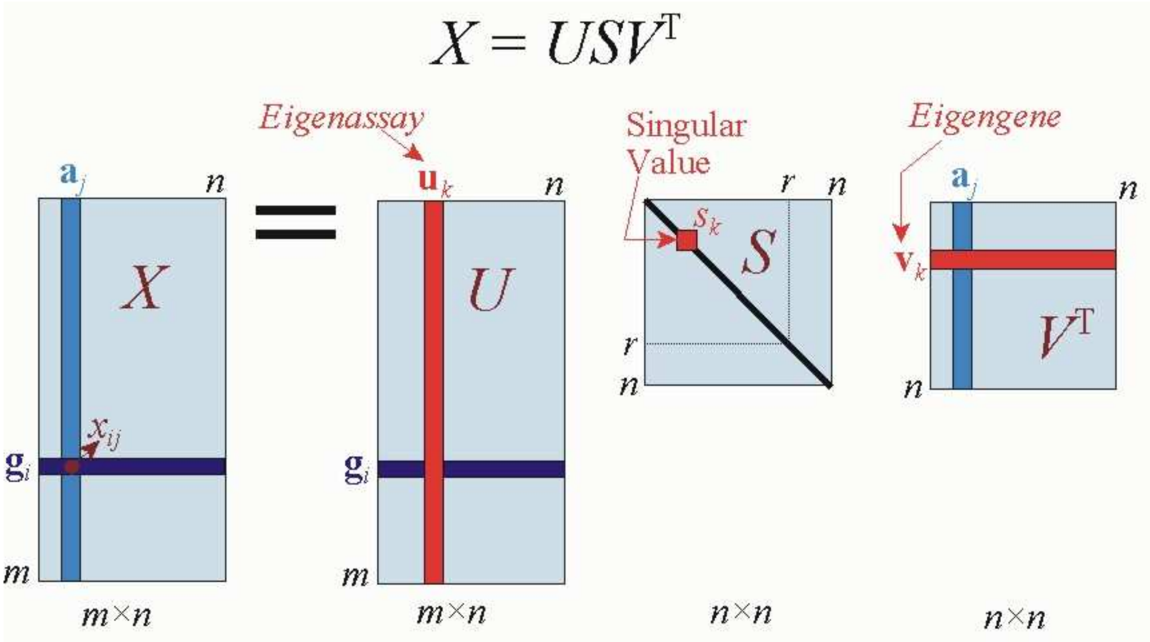


Figure 2.1: Graphical depiction of SVD of a matrix X with notations adopted in this section.

corresponding dimensions are therefore usually neglected in the analysis (i.e. these s_k are set to zero). Use of the SVD in such a way is usually referred to as *dimensionality reduction* of the data (see for example also **Image Processing** in section 2.2).

In diagnostic applications, we may wish to classify tissue samples from individuals with and without a disease. Referring to the definitions in section 2.1, the signal of interest in this case is the assay expression profile $\mathbf{a}_j$. By Equation 2.1, $\mathbf{a}_j$ can be expressed as a linear combination of the eigenassays $\{\mathbf{u}_k\}$:

$$\mathbf{a}_j = \sum_{k=1}^{r} v_{jk} s_k \mathbf{u}_k \tag{2.10}$$

The jth column of V^T, $\mathbf{a}'_j$ (see Fig. 2.1), contains the coordinates of the jth assay in the coordinate system (basis) of the scaled eigenassays, $s_k\mathbf{u}_k$. The $\mathbf{a}'_j$ capture the expression profiles of the assays in $r \leq n$ dimensions, which is always fewer than the m dimensions of the original expression profiles $\mathbf{a}_j$. So, in contrast to gene expression vectors, SVD can generally reduce the dimensionality of the assay expression profiles without neglecting dimensions with small singular values.

As hinted on above, analysis of the spectrum formed by the singular values s_k can lead to the determination that fewer than n components capture the essential features in the data, a topic discussed in more detail below in the section on the visualization of the SVD (section 2.3.1). In the literature the number of components that results from such an analysis is sometimes associated with the number of underlying biological processes that give rise to the patterns in the data. It is then of interest to ascribe biological meaning to the significant eigenassays (in the case of diagnostic applications), or eigengenes (in the case of systems biology applications). For example, in diagnostic applications one eigenassay might correspond to a characteristic expression profile of healthy tissue whereas a second eigenassay might correspond to a characteristic expression profile of diseased tissue. In a systems biology application with time series expression data, two eigengenes might correspond to (roughly) average expression profiles of two different groups of genes involved in two different biological processes. And even though individual components (eigenassays/eigengenes) may not necessarily be biologically meaningful on their own, biologically meaningful signals might be found in two or higher dimensional SVD subspaces (see, e.g., **small-angle scattering** in section 2.2). In the context of describing scatter plots in section 2.3.1, we discuss the application of SVD to the problem of grouping genes by expression vector, and grouping assays by expression profile. This discussion will also touch on the topic of searching for biologically meaningful signals. Sometimes it might not be possible to resolve gene groups, either because there really is no such structure in expression space, or because it has been ’washed out’ by noise in the data. In such cases it might still be of interest to identify the underlying gene expression patterns through the eigengenes, and the expression subspace these patterns span. This is a case where the utility of the SVD distinguishes itself from the typically used clustering techniques (see also section 2.3.2). Finally we discuss some published examples of gene expression analysis using SVD, and a couple of SVD-based gene grouping methods (section 2.3.3).

2.3.1 Visualization of the SVD and the Matrices S, V^T and U

Visualization is central to understanding the results of application of SVD to gene expression data. For example, Figure 2.2 illustrates plots that are derived from applying SVD to Cho et al.’s budding yeast cell-cycle data set [Cho et al., 1998]. In the experiment, roughly 6200 yeast genes were monitored for 17 time points taken at ten-minute intervals. To perform the SVD, the data were pre-processed by replacing each measurement with its logarithm, and normalizing each gene’s expression vector to have zero mean and unit standard deviation. In addition, an autocorrelation filter was applied (see chapter 4) to filter out ~3200 genes that showed primarily random fluctuations in their expression profiles. The plots in Fig. 2.2 reveal interesting patterns in the data that we may wish to investigate further: plot a) shows a leveling off of the relative variance after the first few components; b) shows a pattern in the first eigengene primarily resembling a steady decrease, or decay; plots c) and d) show patterns with cyclic structures in the second and third eigengenes.

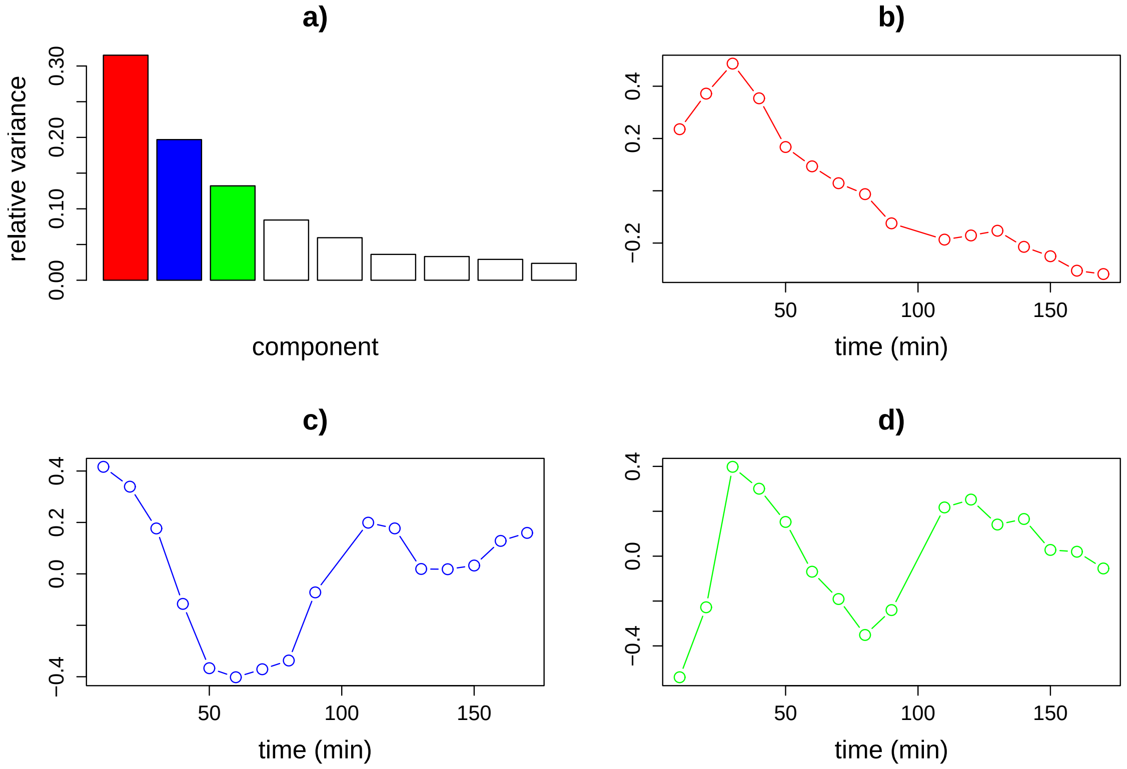


Figure 2.2: Visualization of the SVD of the Cho et al. [Cho et al., 1998] yeast cell-cycle gene expression data. Plots of relative variance (a); and the first (b), second (c) and third (d) eigengenes are shown. The methods of visualization employed in each panel are described in section 2.3.1. These data inspired our choice of the sine and exponential patterns for the synthetic data of section 2.3.1.

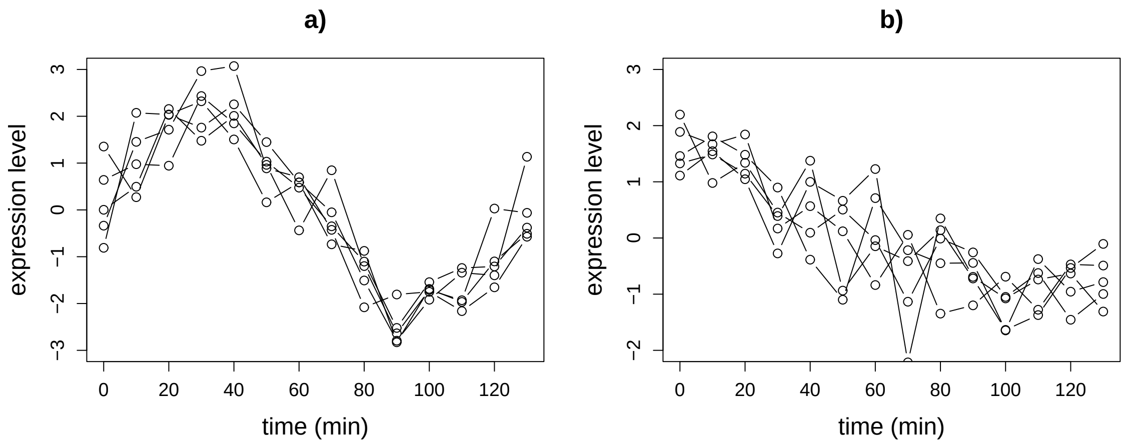


Figure 2.3: Gene expression vectors from the synthetic data set. Overlays of a) five noisy sine wave genes and b) five noisy exponential genes.

A Synthetically Generated Example Data Set

To aid our discussion of visualization, we use a synthetic time series data set with 14 sequential expression level assays (columns of X) of 2000 genes (rows of X). Use of a synthetic data set enables us to provide simple illustrations that can serve as a foundation for understanding the more complex patterns that arise in real gene expression data. Genes in our data set have one of three kinds of expression vector, inspired by experimentally observed patterns in the Cho et al. cell-cycle data: 1) noise (1600 genes); 2) noisy sine pattern (200 genes); or 3) noisy exponential pattern (200 genes). Noise for all three groups of genes was modeled by sampling from a normal distribution with zero mean and standard deviation 0.5. The sine pattern has the functional form $a\,sin(2\pi t/140)$, and the exponential pattern the form $be^{-t/100}$. a was sampled from the uniform distribution over the interval (1.5,3) and b was sampled from the uniform distribution over (4,8). t contains 14 time points covering one period of the sine wave. In analogy with the cell-cycle data, the time points can be thought of as samples taken at 10 min intervals starting at t=0. Each (synthetic) gene expression vector was centered to have a mean of zero. Figure 2.3 depicts genes of type 2) and 3).

2.3.1.1 Singular value spectrum

The diagonal values of S (i.e., s_k) make up the singular value spectrum, which is easily visualized by simply plotting the values. A singular value indicates the importance of a SVD mode in terms of the amount of variance in the data it explains. More specifically, the square of each singular value denotes the variance in the data explained by the corresponding singular vector. The relative variances $s_k^2 / \sum_i s_i^2$ are often plotted (see Fig. 2.4 a) and Fig. 2.2 a)). Cattell has referred to these kinds of plots as *scree plots* [Cattell, 1966] and proposed to use them as a graphical method to decide on the significant components. If the original variables are linear combinations of a smaller number of underlying variables, combined with some low-level noise, the plot will tend to drop sharply for first few singular values associated with the underlying variables and then much more slowly for the remaining singular values, causing an 'elbow' in the plot. Components (eigenassays and eigengenes) whose singular values plot to the right of such an elbow are ignored because they are assumed to be mainly due to noise. In Figure 2.4 a) such an elbow is clearly visible at component 3, as one might expect because only two linearly independent signals (except for the noise genes), the sine and exponential patterns, are present in the data[2].

Other heuristic approaches for deciding on the significant SVD components for a data set have been proposed. One approach is to require the cumulative relative variance of the selected components to be larger than a certain threshold which is usually dependent upon the dimensionality of the data. For gene expression time series data with around a dozen assays it has been found that the first 2 or 3 components usually capture 70% of the variance in the data [Holter et al., 2000]. In the cell-cycle data of Cho et al., for example (see Fig. 2.2 a)), the first three components capture close to 70% of the variance. For our synthetic example data set, the first two components capture about 64% of the total variance in the data (Fig. 2.2 a). If we re-construct the data matrix X for the synthetic example data by using the first two components, we would obtain $X^{(2)}$ (the best rank-2 approximation of X), which would account for 64% of the variance in the data. An alternative approach for component selection was proposed by Everitt and Dunn [Everitt and Dunn, 2001]. They suggest a threshold for the variance captured by the individual components. In their approach, a component is selected as significant if its relative variance is larger than $0.7/r$, where r is again the rank of matrix X [Everitt and Dunn, 2001]. For our example data set this threshold is $th = 0.7/14 = 0.05$, which again selects the first two components as significant. For the yeast cell-cycle data set, with a threshold on the variance of $th = 0.7/17 \approx 0.04$, this approach would select the first 5 components, illustrating that these heuristic approaches for component selection can lead to different results and can not be taken as last truths.

2.3.1.2 Eigengenes

When assays correspond to samplings of an ordinal or continuous variable (e.g., time; radiation dose; toxin concentration), a plot of the elements of the eigengenes $\{\mathbf{v}_k\}$ may reveal recognizable patterns. In our synthetic data set, the first two eigengenes show an obvious cyclic structure (Figs. 2.4 b, c; see also eigenvectors 2 and 3 in Fig. 2.2 for the yeast cell-cycle data). Neither eigengene

[2] Also note that for a real data set, as the yeast cell cycle data set in Figure 2.2 a), such a clear 'elbow' is not visible in the singular value spectrum.

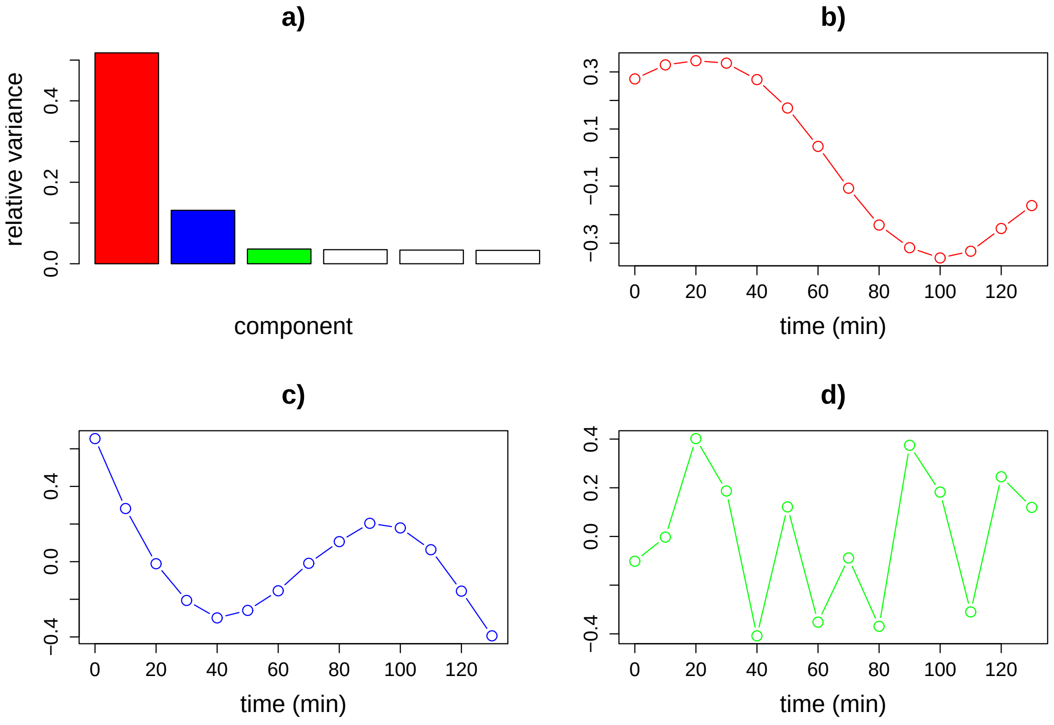


Figure 2.4: Visualization of the SVD of the synthetic data matrix. a) Singular value spectrum (relative variance plot). The first two singular values account for 64% of the variance. The first (b), second (c), and third (d) eigengenes are plotted vs. time (assays) in the remaining panels. The first two eigengenes capture the signals of the sin and exponential patterns completely. There is no signal in the third eigengene and it only represents noise.

is exactly like the underlying sine or exponential pattern, as the two patterns are not orthogonal. However, each original pattern in the data is closely approximated by a linear combination of the first two eigengenes. When assays correspond to discrete experimental conditions (e.g., mutational varieties; tissue types; distinct individuals), visualization schemes are similar to those described below for eigenassays. When the jth element of eigengene *k* is of large-magnitude, the jth assay is understood to contribute relatively strongly to the overall variance of eigenassay *k*, a property that may be used for associating a group of assays.

2.3.1.3 Eigenassays

Alter et al. [Alter et al., 2000] have visualized eigenassays $\{\mathbf{u}_k\}$ resulting from SVD analysis of cell-cycle data by adapting a previously developed color-coding scheme for visualization of gene expression data matrices [Eisen et al., 1998]. For visualization, individual elements of *U* are displayed as rectangular pixels in an image, and color-coded using green for negative values, and red for positive values, the intensity being correlated with the magnitude. The rows of matrix *U* can be sorted using correlation with the eigengenes. In Alter et al.'s study, this scheme sorted the genes by the phase of their periodic pattern. The information communicated in such visualization bears some similarity to visualization using scatter plots, with the advantage that the table-like display enables gene labels to be displayed along with the eigenassays, and the disadvantage that differences among the genes can only be visualized in one dimension.

2.3.1.4 Visualization of Genes and Assays with Scatter Plots

Visualization of structure in high-dimensional data requires display of the data in a one, two, or three-dimensional subspace. SVD identifies the subspaces in which the data varies the most. Even though our discussion here is about visualization in subspaces obtained by SVD, the illustrated visualization techniques are general and can in most cases be applied for visualization in other subspaces (see section further reading and resources for techniques that use other criteria for subspace selection). For gene expression analysis applications, we may want to classify samples in a diagnostic study, or classify genes in a systems biology study. Projection of data into SVD subspaces and visualization with scatter plots can reveal structures in the data that may be used for classification. Here we discuss the visualization of features that may help to distinguish gene groups by their expression vectors. Analogous methods are used to distinguish groups of assays by expression profile. We discuss two different sources of gene "coordinates" for scatter plots: projections of the expression vector onto eigengenes, and correlations of the expression vector with eigengenes.

Projection and correlation scatter plots for gene expression vectors Projection scatter plot coordinates q_{ik} for expression vector $\mathbf{g}_i$ projected on eigengene $\mathbf{v}_k$ are calculated as

$$q_{ik} = \mathbf{g}_i \mathbf{v}_k \tag{2.11}$$

The SVD of X readily allows computation of these coordinates using the equation $XV = US$, so that

$$q_{ik} = (US)_{ik} \tag{2.12}$$

As we noted before, the projection of the $\mathbf{g}_i$ onto the eigengenes $\mathbf{v}_k$ in V represents a rotation of the $\mathbf{g}_i$ from the original basis.

The projection of gene expression vectors from our example data onto the first two eigengenes reveals the a priori known structure of the data (Fig. 2.5 a)). The groups of the 200 sine wave genes (bottom right cluster), and the 200 exponential decay genes (top right cluster) are clearly separated from each other and from the 1600 pure noise genes, which cluster about the origin.

Correlation scatter plots may be obtained by calculating the Pearson correlation coefficient of each gene's expression vector with the eigengenes:

$$r_{ik} = \frac{\delta \mathbf{g}_i \delta \mathbf{v}_k}{|\delta \mathbf{g}_i|\,|\delta \mathbf{v}_k|} \tag{2.13}$$

where r_{ik} denotes the correlation coefficient of the expression vector $\mathbf{g}_i$ with eigengene $\mathbf{v}_k$, $\delta \mathbf{g}_i$ the mean-centered $\mathbf{g}_i$, and $\delta \mathbf{v_k}$ is the mean-centered $\mathbf{v}_k$. The normalization by the lengths of the vectors $|\delta \mathbf{g}_i|$ and $|\delta \mathbf{v}_k|$[3] leads to $-1 \leq r_{ik} \leq 1$. Note that if each $\mathbf{g}_i$ is pre-processed to have zero mean and unit norm,

$$r_{ik} = q_{ik} = (US)_{ik} \tag{2.14}$$

and it follows that the correlation scatter plot is equivalent to the projection scatter plot ($\mathbf{g}_i = \delta \mathbf{g_i}$ implies $\mathbf{v_k} = \delta \mathbf{v_k}$; and $|\delta \mathbf{g_i}| = |\delta \mathbf{v_k}| = 1$). In the projection scatter plot, genes with a relatively high-magnitude coordinate on the k-axis contribute relatively strongly to the variance of the kth eigengene in the data set. The farther a gene lies away from the origin, the stronger the contribution of that gene is to the overall variance accounted for by the subspace. In the correlation scatter plot, genes with a relatively high-magnitude coordinate on the k-axis have expression vectors that are relatively highly correlated with the kth eigengene.

Due to the normalization in correlation scatter plots, genes with similar patterns in their expression vectors, but with different amplitudes, can appear to cluster more tightly in a correlation scatter plot than in a projection scatter plot. Genes that correlate well with the eigengenes lie near the perimeter, a property that can be used in algorithms that seek to identify genes that are highly

[3] Which are equal the sample standard deviations of the respective mean centered vectors.

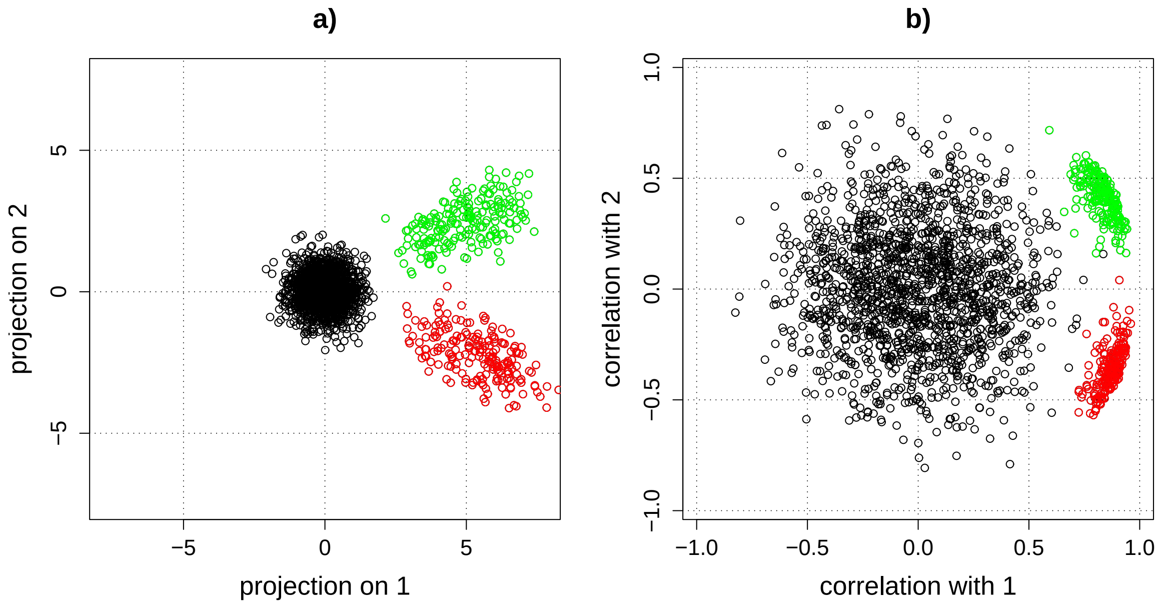


Figure 2.5: SVD scatter plots. Genes from our synthetic example data set are displayed in a) a projection scatter plot; and b) a correlation scatter plot. The bottom right cluster (red) corresponds to sine wave genes, and the top right cluster (green) corresponds to exponential decay genes. The cluster of genes around the origin corresponds to the noise-only genes.

associated with a subspace. At the same time, low-amplitude noise genes can appear to be magnified in a correlation scatter plot. For our example data, the sine wave and exponential gene clusters are relatively tightened, the scatter of the noise genes appears to be increased, and the separation between signal and noise genes is decreased for the correlation vs. the projection scatter plot (Fig. 2.5). The projection scatter plot (Fig. 2.5 a) illustrates how SVD may be used to aid in detection of biologically meaningful signals. In this case, the position $(q_1^{c_i}, q_2^{c_i})$ of any cluster c_i's center may be used to construct the cluster's expression vector $\mathbf{g}^{c_i}$ from the right singular vectors:

$$\mathbf{g^{c_i}} = q_1^{c_i}\mathbf{v}_1 + q_2^{c_i}\mathbf{v_2} \tag{2.15}$$

If the first and second singular vectors are biologically meaningful in and of themselves, the cluster centers will lie directly on the axes of the plot. This requires that the average expression patterns of the cluster genes are uncorrelated. For our synthetic data, the original sine and exponential patterns have a correlation larger than zero, therefore they have to be represented as linear combinations of two uncorrelated singular vectors (eigengenes).

SVD and related methods are particularly valuable analysis methods when the distribution of genes is more complicated than the simple cluster-like distributions in our example data: for instance, SVD can be used to characterize ring-like distributions of genes such as are observed in scatter plots of cell-cycle gene expression data [Alter et al., 2000, Holter et al., 2000] (see section 2.3.3).

Scatter plots of assays Assays can be visualized in scatter plots using methods analogous to those used for genes. Coordinates for projection scatter plots are obtained by taking the dot products $q_{kj} = \mathbf{u}_k \mathbf{a_j}$ of eigenassays $\mathbf{u}_k$with expression profiles $\mathbf{a}_j$. These projections are again readily obtained from SVD as $q_{kj} = (U^T X)_{kj} = (SV^T)_{kj}$. Coordinates for correlation scatter plots are obtained by calculating the Pearson correlation coefficient $\delta\mathbf{a}_j \delta\mathbf{u}_k / \left|\delta\mathbf{a}_j\right| \left|\delta\mathbf{u}_k\right|$. Such plots are useful for visualizing diagnostic data, e.g., distinguishing groups of individuals according to expression profiles. Alter et al. used such a technique to visualize cell-cycle assays [Alter et al., 2000], and were able to associate individual assays with different phases of the cell cycle.

2.3.2 Detection of weak expression patterns

As noise levels in the data increase, it is increasingly difficult to obtain separation of gene groups in scatter plots. In such cases SVD may still be able to detect weak patterns in the data that may be associated with biological effects. In this respect SVD and related methods provide information that is unique from commonly used clustering techniques.

We will use an example to illustrate the ability of SVD to detect patterns in gene expression data even though the individual genes may not clearly separate in expression space (as visualized in a 2-dimensional scatter plot). A data matrix was generated using two kinds of expression vector: 1000 genes exhibiting a sine pattern, $sin(2\pi t/140)$, with added noise sampled from a normal distribution of zero mean and standard deviation 1.5; and 1000 genes with just noise sampled from the same distribution. Upon application of SVD, we find that the first eigengene shows a coherent sine pattern (Fig. 2.6 a). The second eigengene is dominated by high-frequency components that come from the noise (Fig. 2.6 b), and the singular value spectrum is basically flat after the first singular value (Fig. 2.6 c), suggesting (as we know a priori) that there is only one interesting signal in the data. Even though the SVD detected the cyclic pattern in the first eigengene, the sine wave and noise-only genes are not clearly separated in the SVD eigengene projection scatter plot (Fig. 2.6 d).

2.3.3 Examples from the literature

Cell-cycle gene expression data display strikingly simple patterns when analyzed using SVD. Here we discuss two different studies that, despite having used data from different experiments and different pre-processing methods, have produced similar results [Alter et al., 2000,Holter et al., 2000].

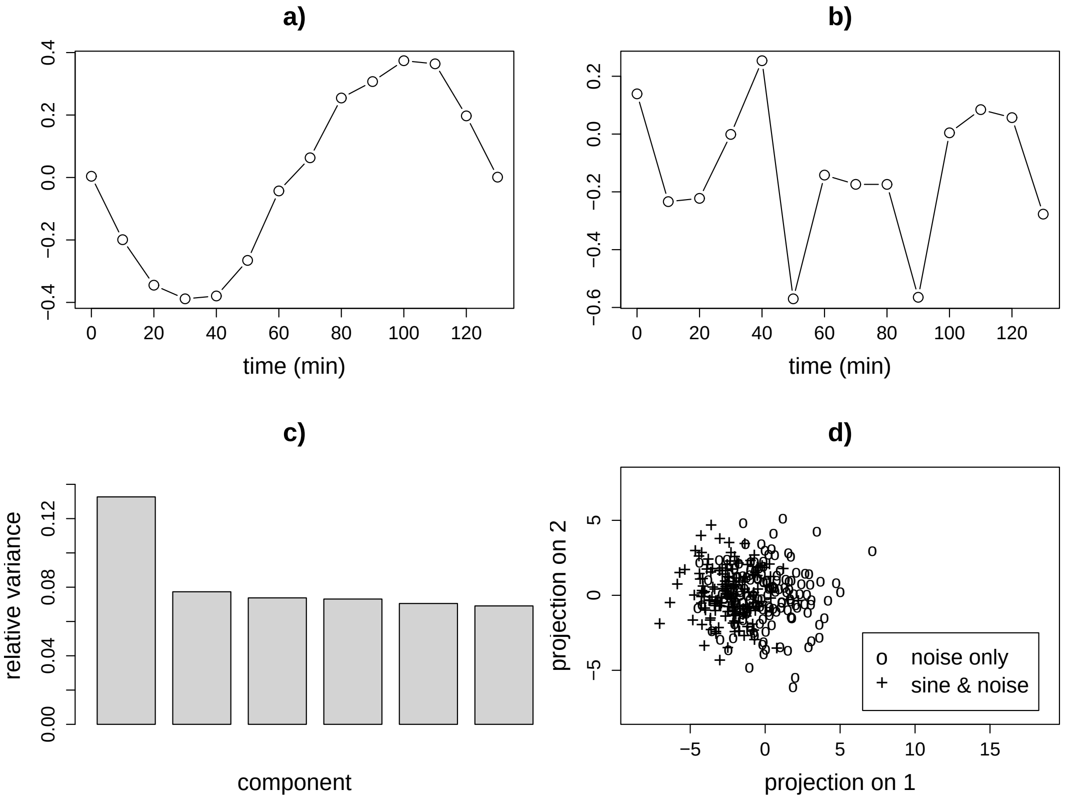


Figure 2.6: SVD-based detection of weak signals. a) A plot of the first eigengene shows the structure of the weak sine wave signal that contributes to the expression vector for half of the genes. b) The second eigengene resembles noise. c) A relative variance plot for the first six singular values shows a flat spectrum after the first singular value. d) The signal and noise genes are not separated in an eigengene scatter plot of 150 of the signal genes, and 150 of the noise-only genes.

Both studies found cyclic patterns for the first two eigengenes (see Fig. 4.1 for Holter et al.), and, in two-dimensional correlation scatter plots, previously identified cell cycle genes tended to plot towards the perimeter of a disc (see Fig. 4.2). Alter et al. used information in SVD correlation scatter plots to obtain the result that 641 of the 784 cell-cycle genes identified in Spellman et al. [Spellman et al., 1998] are significantly associated with the first two eigengenes. Holter et al. displayed previously identified cell-cycle gene clusters in scatter plots, revealing that cell-cycle genes were relatively uniformly distributed in a ring-like feature around the perimeter, leading Holter et al. to suggest that cell-cycle gene regulation may be a more continuous process than had been implied by the previous application of clustering algorithms (see Fig. 4.2). Raychaudhiri et al.'s PCA study of yeast sporulation time series data [Raychaudhuri et al., 2000] is the first example of application of either PCA or SVD to microarray analysis. In this study, over 90% of the variance in the data was explained by the first two components of the PCA. The first principal component contained a strong steady-state signal. Projection scatter plots were used in an attempt to visualize previously identified gene groups, and to look for structures in the data that would indicate separation of genes into groups. No clear structures were visible that indicated any separation of genes in scatter plots. Holter et al.'s more recent SVD analysis of yeast sporulation data [Holter et al., 2000] made use of a different pre-processing scheme from that of Raychaudhuri et al. The crucial difference is that the rows and columns of X in Holter et al.'s study were iteratively centered and normalized, i.e., the mean value of the (row, column) was subtracted from each element in the (row, column), and each element was divided by the standard deviation. In Holter et al.'s analysis, the first two eigengenes were found to account for over 60% of the variance for yeast sporulation data. The first two eigengenes were significantly different from those of Raychaudhuri et al., with no steady-state signal, and, most notably, structure indicating separation of gene groups was visible in the data. Below we discuss the discrepancy between these analyses of yeast sporulation data.

2.4 Discussion

As illustrated in section 2.3.2, an important capability distinguishing SVD and related methods from other analysis methods is the ability to detect weak signals in the data. Even when the structure of the data does not allow separation of data points, causing clustering algorithms to fail, it may be possible to detect biologically meaningful patterns. As an example of practical use of this kind of SVD-based analysis, it may be possible to detect whether the expression profile of a tissue culture changes in response to radiation dose, even when it is difficult to clearly separate the specific genes that change their expression in response to radiation dose from other genes due to noise in expression profiles.

SVD allows us to obtain the dimension of the Euclidean space in which the data can be embedded[4], which is the rank r of matrix X. As the number of genes m is generally (at least presently) greater than the number of assays n, the matrix V^T generally yields a representation of the assay expression profiles using a reduced number of variables. When $r < n$, the matrix U yields a representation of the gene expression vectors using a reduced number of variables. Although this property of the SVD is commonly referred to as dimensionality reduction, we note that any reconstruction of the original data requires generation of an m × n matrix, and thus requires a mapping that involves all of the original dimensions. Given the noise present in real data, in practice the rank of matrix X will always be n, leading to no direct dimensionality reduction for the gene expression vectors. It may be possible, however, to detect the true rank r by ignoring certain components (typically the lower order ones), thereby reducing the number of variables required to represent the gene expression vectors. Previous analyses of gene expression data have found that 2 to 3 components capture much of the significant expression change in the data [Alter et al., 2000, Holter et al., 2000].

Current thoughts about use of SVD/PCA for gene expression analysis include application of SVD as pre-processing for clustering. Clustering algorithms can be applied using, e.g., the coordinates calculated for scatter plots instead of the original data points. Yeung and Ruzzo have characterized the effectiveness of gene clustering both with and without pre-processing using PCA [Yeung and Ruzzo, 2001]. The pre-processing consisted of using PCA to select only the highest-variance principal components, thereby choosing a reduced number of variables for each gene's expression vector. The reduced variable sets were used as inputs to clustering algorithms. For the specific clustering algorithms and data tested, Yeung and Ruzzo report better results without pre-processing. However, the specific data sets and clustering algorithms tested, and the sole focus on gene clustering limit the implications of the results. For example, when grouping assays is the objective, using $\{a'_j\}$ instead of $\{a_j\}$ (see section 2.3; Fig. 2.1) enables use of a significantly reduced number of variables (r vs. m) that account for all of the variance in the data. The resulting reduction of variables trivially decreases the compute time for clustering of assays, and may even result in higher-quality clusters. Hence, at least for assay clustering the results of Yeung and Ruzzo can't be correct.

In section 2.3.3 we discussed how, rather than separating into well-defined groups, cell-cycle genes tend to be more continuously distributed in SVD projections. For instance, when plotting the correlations of genes with the first two right singular vectors, cell-cycle genes appear to be relatively uniformly distributed about a ring. This structure suggests that, rather than using a classification method that groups genes according to their co-location in the neighborhood of a point (e.g., k-means clustering), one should choose a classification method appropriate for dealing with

[4] E.g. the dimension of data points distributed on a sphere is two, the dimension of the Euclidean space in which the data can be embedded is three. The latter is the dimesnionality that SVD provides.

ring-like distributions. Previous cell-cycle analyses therefore illustrate the fact that one important use of SVD is to aid in selection of appropriate classification methods by investigation of the dimensionality of the data. SVD analysis expresses signals in the data as linear combinations of orthogonal signals. A common misconception is that SVD can only reveal information about underlying signals that are orthogonal. In fact, SVD may be used to detect underlying signals that are not orthogonal, as can be seen from our synthetic example data set with sine and exponential patterns (see Figures 2.5; see also small-angle scattering in section 2.2).

In this chapter we have concentrated on conveying a general understanding of the application of SVD analysis to gene expression data. Here we briefly mention several specific SVD-based methods that have been published for use in gene expression analysis. For gene grouping, the *gene shaving* algorithm of Hastie et al. [Hastie et al., 2000] and SVDMAN by Wall et al. [Wall et al., 2001] are available. An important feature to note about both gene shaving and SVDMAN is that each gene may be a member of more than one group (e.g. cluster for clustering algorithms). For an evaluation of the data, SVDMAN uses SVD-based interpolation of deleted data to detect sampling problems when the assays correspond to a sampling of an ordinal or continuous variable (e.g., time series data). A program called SVDimpute [Troyanskaya et al., 2001] implements an SVD-based algorithm for imputing missing values in gene expression data. Holter et al. have developed an SVD-based method for analysis of time series expression data [Holter et al., 2001]. The algorithm estimates a time translation matrix that describes evolution of the expression data in a linear model. Yeung et al. have also made use of SVD in a method for reverse engineering linearly coupled models of gene networks [Yeung et al., 2002].

It is important to note that application of SVD and PCA to gene expression analysis is relatively recent, and that methods are currently evolving. There is no theory that dictates how to perform SVD-based gene expression analysis, and there is no software package to date that implements an automated general-purpose gene expression analysis. The detailed path of any given analysis thus depends on what specific scientific questions are being addressed. Presently, gene expression analysis in general tends to consist of iterative applications of interactively performed analysis methods. As new inventions emerge, and more techniques and insights are obtained from other disciplines, we mark progress towards the goal of an integrated, theoretically sound approach to gene expression analysis; much remains to be accomplished, however, before we reach that goal.

2.5 Further Reading and Resources

The book of Jolliffe [Jolliffe, 1986] is a fairly comprehensive reference on PCA. It gives interpretations of PCA and provides many example applications, with connections to and distinctions from other techniques such as correspondence analysis and factor analysis. For more details on the mathematics and computation of SVD, good references are [Golub and Van Loan, 1996, Strang, 1998, Berry, 1992, Jessup and Sorensen, 1994]. SVDPACKC has been developed to compute the SVD algorithm [Berry et al., 1993][5]. SVD is also used in the solution of unconstrained linear least squares problems, matrix rank estimation, and canonical correlation analysis [Berry, 1992]. Applications of PCA and/or SVD to gene expression data have been published in [Alter et al., 2000, Holter et al., 2000, Holter et al., 2001, Raychaudhuri et al., 2000, Troyanskaya et al., 2001, Yeung and Ruzzo, 2001, Yeung et al., 2002]. Many of the aspects of these studies were discussed in sections 2.3.3 and 2.4. In addition, SVDMAN [Wall et al., 2001] and gene shaving [Hastie et al., 2000] are published SVD-based grouping algorithms; SVDMAN is free software available at http://home.lanl.gov/svdman. Knudsen illustrates some of the uses of PCA for visualization of gene expression data [Knudsen, 2002].

Everitt, Landau and Leese [Everitt et al., 2001] present PCA as a special case of Projection Pursuit [Friedman and Tukey, 1974]. Projection Pursuit, which in general attempts to find an "interesting projection" for the data, is also related to Independent Component Analysis (ICA) [Hyvarinen, 1999]. As the name implies, ICA attempts to find a linear transformation (non-linear generalizations are possible) of the data so that the derived components are as much as possible statistically independent from each other. Hyvärinen provides a discussion of ICA and how it relates to PCA and Projection Pursuit [Hyvarinen, 1999]. Liebermeister has applied ICA to gene expression data [Liebermeister, 2002]. Other techniques that are related to PCA and SVD for visualization of data are Multidimensional Scaling [Borg and Groenen, 1997] and Self-Organizing Maps (SOM) [Kohonen, 2001]. Both of these techniques use non-linear mappings of the data to find lower-dimensional representations. SOM's have been applied to gene expression data in [Tamayo et al., 1999]. There are also non-linear generalizations of PCA [Jolliffe, 1986, Schölkopf et al., 1996].

[5] Some resources on SVD can also be found on the Web, see for example the following URL's: http://www.cs.ut.ee/~toomas_l/linalg/; http://www.lapeth.ethz.ch/~david/diss/node10.html; and http://www.stanford.edu/class/cs205/notes/book/book.html.

Chapter 3

SVD identifies different Modes of Response to Virus Infection[1]

3.1 Introduction

3.1.1 Biological Model

Global gene expression analysis using DNA gene chip technology makes it possible to simultaneously monitor the expression levels of large numbers of mRNAs in cells [Duggan et al., 1999, Schena et al., 1995]. One area where gene chip analysis has been useful is in studying host-pathogen interactions. Gene chip analysis allows for the comparison of gene expression levels in infected and uninfected cells. One pathogen that has been studied by this approach is human cytomegalovirus (HCMV), a member of the herpesvirus subfamily betaherpesvirinae. HCMV causes life-threatening disease in immunologically immature and immunocompromised people, including neonates, AIDS patients and allogenic transplant recipients [Challacombe et al., 2004].

Previous studies of global host gene expression using DNA microarrays have shown that HCMV infection dramatically changes the gene expression profile of the host cell [Challacombe et al., 2004]. HCMV infection alters the expression of numerous host cell genes, including genes that regulate cell cycle progression, cellular proliferation, cell adhesion, and genes encoding transcription factors. Human cells respond to HCMV infection by altering transcription in an attempt to antagonize viral replication and spread.

[1] Work that was published as part of [Challacombe et al., 2004].

3.1.2 Methods of handling data

To go from raw gene expression data to meaningful results generally involves normalization, filtering, and analysis to identify patterns in expression level data. With Affymetrix microarray experiments, the raw data consist of probe pair intensities. The gene expression level is typically computed using a statistic that captures the response characteristic of a specific probe set. Many different commercial and free software packages can perform normalization and expression analysis of oligonucleotide arrays. A few examples are DNA-Chip Analyzer (dChip) [Li and Wong, 2001a], Affymetrix's GeneChip software [Affymetrix, 1999], GeneSpring (http://www.silicongenetics.com), Cluster and TreeView [Eisen et al., 1998]. In this study, we compared the group of human genes that responded to HCMV infection in a previous study using GeneChip and a fold change approach [Browne et al., 2001], to two clusters of co-expressed genes that we identified using dChip and Singular Value Decomposition (SVD) analysis. The first cluster contained some of the genes identified in the previous study [Browne et al., 2001], while nearly all genes in the second cluster were not identified previously.

3.1.3 Materials and Methods

We analyzed gene expression time course data (from Affymetrix CEL files; NCBI Gene Expression Omnibus accession GSE675 available at http://www.ncbi.nlm.nih.gov/geo/) obtained after HCMV infection of human fibroblast cells by [Browne et al., 2001]. dChip [Li and Wong, 2001a, Li and Wong, 2001b] was used to normalize the intensities of the array data and estimate the expression levels. SVD [Wall et al., 2003, Golub and Van Loan, 1996] was employed to identify and visualize the two dimensional subspace that captured most of the variance in the expression data. In this subspace, two clusters of co-expressed genes were identified. We annotated the genes comprising these clusters and grouped them into functional categories.

3.1.3.1 Expression Level Estimation

One key issue in expression level estimation of oligonucleotide chips is the way that probe-specific effects are handled. Affymetrix's GeneChip uses the average difference of the perfect match (PM) and mismatch (MM) probes as an expression index for the target gene. However, even using MM intensities as controls, the expression levels of the different probe pairs in a probe set are still highly variable [Li and Wong, 2001a]. dChip accounts for probe-specific effects in the computation of expression levels by using a probe-sensitivity index to capture the response characteristic of a specific probe pair, and by calculating model-based expression indices [Li and Wong, 2001a].

3.1.3.2 Expression Level Data Analysis

Singular Value Decomposition (SVD) is a standard technique for dimensionality reduction and interpretation of data [Golub and Van Loan, 1996, Wall et al., 2003] (see chapter 2 in this dissertation). When applied to a gene expression matrix consisting of the expression levels of m genes measured at n time points (assays), SVD can be viewed as a linear transformation of the expression data from an m x n space to a number of characteristic modes that describe the temporal patterns of gene expression. The SVD analysis of the dChip normalized data was performed with the statistical software package R[2] [R Development Core Team, 2004]. The R svd function provides an interface to the LINPACK routine DSVDC. Prior to performing SVD, the data was log transformed and for each gene its transcriptional response vector was centered by subtracting its mean and then standardized to unit variance. Following SVD analysis, we calculated the correlations of the transcriptional response vectors with the first two modes, then visualized the correlations in a scatter plot (see chapter 2 for an explanation of the scatter plot). The plot indicates two distinct clusters of genes, one correlated with each mode. We identified the set of genes in each cluster by visual inspection of the correlation plot and by manually identifying a boundary around the cluster based on the density of genes. The genes in our newly identified clusters were ranked by the magnitude of their expression variance and exported as a list to a file, where each gene was identified by its Affymetrix probe set id. Since the genes in a co-expression cluster have similar transcriptional response patterns, the variance is a measure of the amplitude (or magnitude) of the transcriptional response vectors.

3.1.3.3 Annotation Protocol

The gene chip IDs were mapped to GenBank accession numbers and uploaded to the Stanford University's sourceBatchSearch (http://genome-www5.stanford.edu/cgi-bin/SMD/source//sourceBatchSearch) to obtain annotation information for each GenBank accession number.

3.2 Results

3.2.1 Data Normalization

The GeneChip model is additive, and models the expression level of a given gene by the sum of the probe effect and gene effect plus a stochastic component, which represents the measurement error. The dChip model is multiplicative, modeling the expression level of a given gene as the product

[2] Freely available under the GNU public license. R and its plugin BioConductor are becoming widely used in gene expression analysis.

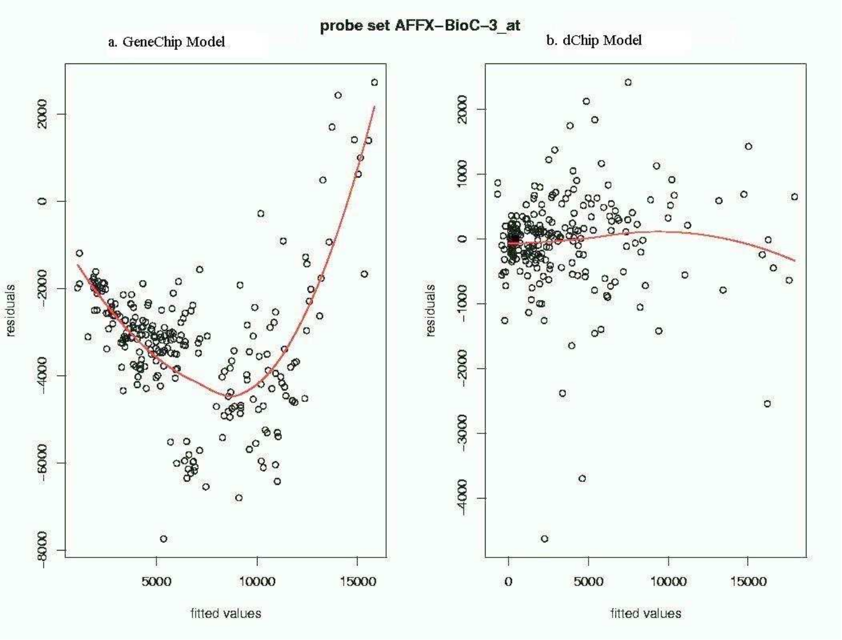


Figure 3.1: Graphs showing the residuals of the statistical models used by GeneChip (a) and dChip (b), applied to probe set AFFX-Bio-C-3_at. The multiplicative model of dChip seems to explain the variation in the data better.

of the probe effect and gene effect plus a stochastic component (measurement error). The fitted values give estimates of the expression levels (probe effect plus gene effect for the additive model and probe effect times gene effect for the multiplicative model). The residuals were calculated as the observed values minus the fitted values. Figure 3.1 shows the plots of the residuals vs. the fitted intensity values for the statistical models used by Affymetrix's GeneChip and dChip applied to the data for the probe set AFFX-Bio-C-3_at. The GeneChip model (Figure 13.1) shows a strong, non-linear dependence of the residuals on the intensity values. The residuals in the dChip model (Figure 3.1b) were smaller (note the different scale on the plots) with a more constant and symmetric spread and far less dependence on the intensity values. This indicates that the multiplicative dChip model explains more of the variation in the expression data and is a better fit to the data.

3.2.2 Main Modes of Host Cell Expression Response to Herpes Infection

SVD analysis of gene expression data usually shows a decreasing singular value spectrum with a leveling off after the first 2-3 modes. The ordering of the modes is determined by high-to-low sorting of the corresponding singular values. The first few modes account for most of the patterns (i.e. variance) in the data, while the rest typically represent noise, which can aid in identifying the most prevalent signals in the data. The reduction of dimensionality provided by SVD analysis

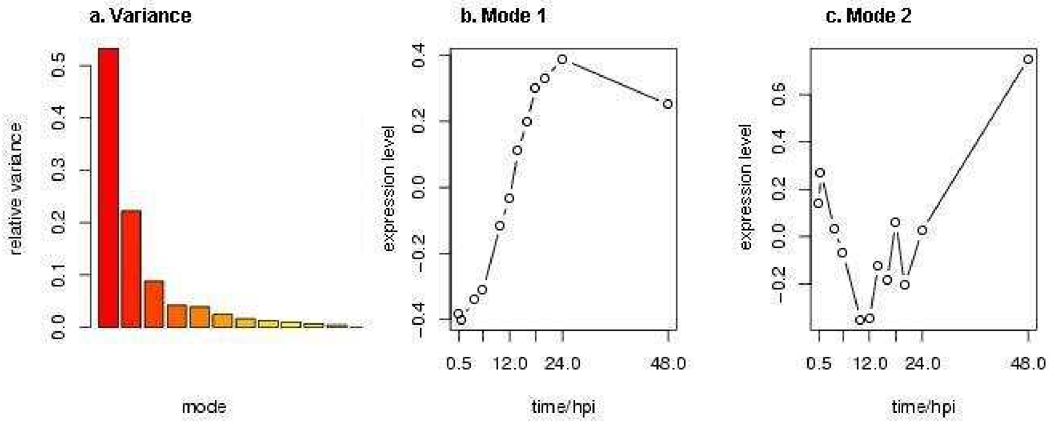


Figure 3.2: The variance captured by each SVD mode for the herpes data is shown in Fig (a). The figure indicates that modes 1 and 2 capture 75% of the variance in the data. Figures (b) and (c) show the expression profiles of modes 1 and 2.

facilitates data visualization, and clusters of genes with similar transcriptional responses might be identified. Figure 3.2 shows the singular value spectrum and first two modes of the dChip modeled herpes data. The expression data consists of 12 time points (12 arrays), representing 0.5, 1, 4, 6, 10, 12, 14, 16, 18, 20, 24 and 48 h after HCMV infection. The first two modes capture 75% of the variance in the data. 3.2b shows the pattern of expression of the first mode over time. This mode contains most of the variance in the data. At 1 h after HCMV infection, the expression pattern of the first mode increases sharply, up to the level at 24 h, and then decreases slightly over the next 24 h. (Note that the decrease at 48 h was based on 1 data point; we didn't have data points between 24 and 48 h, so an artifact in the 48 h array could have affected the results). All genes that are highly correlated with the first mode show a similar transcriptional response to the pattern of mode 1. Orthogonal to the first mode, the second mode (Figure 3.2c) captures most of the remaining variance in the data. The pattern comprising the second mode decreases until about 12 h after infection, then increases and is somewhat higher at 48 h than for the early time points. Genes highly correlated with this mode show similar transcriptional response. Note that whereas genes correlated with mode 1 are up-regulated initially after HCMV infection, genes correlated with mode 2 are down-regulated. This suggests that genes correlated with mode 1 are activated by the host's immune response whereas genes correlated with mode 2 are down-regulated by the virus proteins in an attempt to evade the hosts immune response. The third mode captures less than 10% of the variance of the data (Fig. 3.2a) and only few genes were found to be highly correlated with this mode. We therefore ignore mode 3, as it probably contains mostly noise.

3.2.3 Visualization of Gene Clusters in Two-dimensional Expression Subspace

Figure 3.3a shows a correlation plot of the gene transcriptional response vectors with modes 1 and 2. The closer the genes map to the periphery of the circle with radius 1, the more their transcriptional response vectors are correlated with the first two modes. We identified two regions where the genes cluster densely. Both regions are close to the perimeter of the plot. One cluster is highly correlated with the first mode (right side of the plot in Figure 3.3a). We manually identified a boundary for this cluster (see Fig 3.4a). This cluster 1 boundary contains 1747 genes and Fig 3.4b shows the average transcriptional response of these genes. As cluster 1 genes are highly correlated with mode 1, their average transcriptional response pattern is very similar to the expression profile of mode 1 (Fig 3.2b). The second high density region of genes is highly correlated with the second mode and somewhat (anti-)correlated with the first mode. Figure 3.5a shows the boundary we selected for this cluster 2. Figure 3.5b shows the average transcriptional response pattern for the 462 genes in this cluster. The similarity of this cluster's response to the profile of mode 2 (see Fig 3.2c) is apparent.

Comparison of Results to Previous Study by [Browne et al., 2001] The criterion used by [Browne et al., 2001] for genes to be selected as significantly expressed requires a fold change (ratio) of 3 in expression over a control measurement in two sequential time points[3]. The control measurement was obtained from "mock infected" cells, i.e. cells that were infected without a virus present. [Browne et al., 2001] used Affymetrix's GeneChip software for the normalization and determination of the probe set (i.e. gene) expression levels from the probe expression levels. Figure 3.3b shows the projections of the genes identified as significantly expressed in the study of [Browne et al., 2001] onto the first 2 modes. The center of Fig. 3.3b is sparsely populated and most of the transcriptional response vectors are highly correlated (or anti-correlated) with modes 1 and 2. This indicates that modes 1 and 2 also explain most of the variance of the genes identified as significantly expressed in the previous study by [Browne et al., 2001]. Many of the genes in this group were highly correlated with mode 1 and somewhat fewer were anti-correlated with the first mode. 377 (or 26%) of the genes selected by the fold change approach of [Browne et al., 2001] were within our cluster 1. Genes in the second cluster, highly correlated with the second mode, were not well represented by the group of previously identified genes. Only 15 genes (or 1%) in the second cluster were selected by the original analysis.

[3]

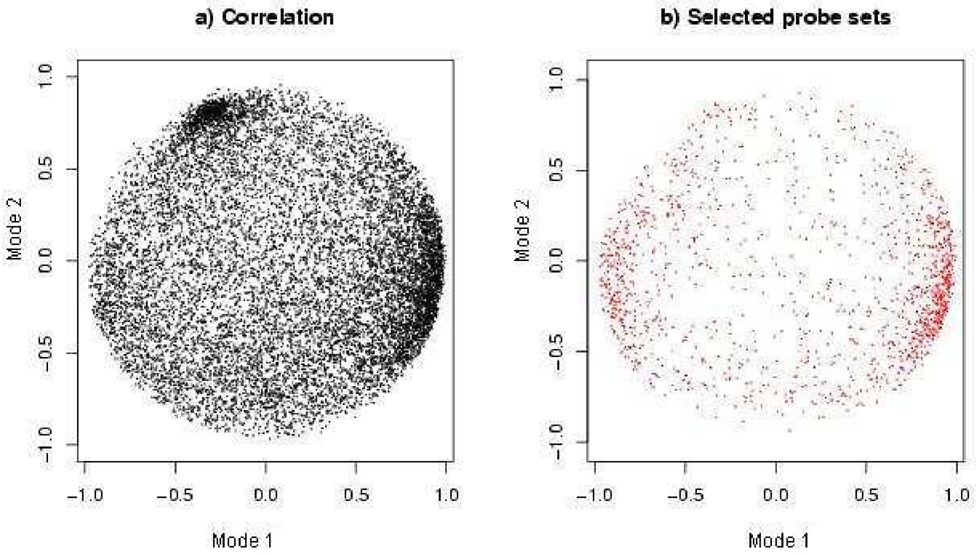


Figure 3.3: Visualization of the gene transcriptional response vectors in a correlation plot with the expression profiles of modes 1 and 2 (a). Two high density regions, or clusters, one correlated with mode 1 and the other with mode 2, are clearly visible. A correlation plot of the genes originally analyzed by [Browne et al., 2001] with the same modes 1 and 2 is shown in (b). Most of the genes identified by [Browne et al., 2001] are either highly correlated with mode 1 or anti-correlated. Few genes from our cluster 2, highly correlated with mode 2, are among the Browne analyzed data.

Statistical Significance of Identified Clusters To assess the statistical significance of the identified clusters, we calculated the expected number of genes found in same sized cluster regions by chance. The cluster boundaries were randomly rotated in the 2-dimensional space of modes 1 and 2. If the rotated cluster-boundary did not overlap with the two identified clusters, the number of genes inside the cluster boundary was counted. The mean and standard deviation of the number of genes inside the boundary was calculated from 100 samples. For the cluster 1 boundary, the mean number of genes found inside the boundary was 337 with a standard deviation of 98 genes. The number of genes that we identified in cluster 1 (1747) was more than 5 times higher than the mean number of genes obtained by chance. For the cluster 2 boundary, we found a mean of 76 genes with a standard deviation of 21; the number of genes identified in cluster 2 (462) was 6 times higher.

3.2.4 Biological functions of genes in clusters 1 and 2

We manually analyzed the tab-delimited SourceSearch files, which contained the annotated genes comprising each cluster, looking for genes that participate in biological processes relevant to the

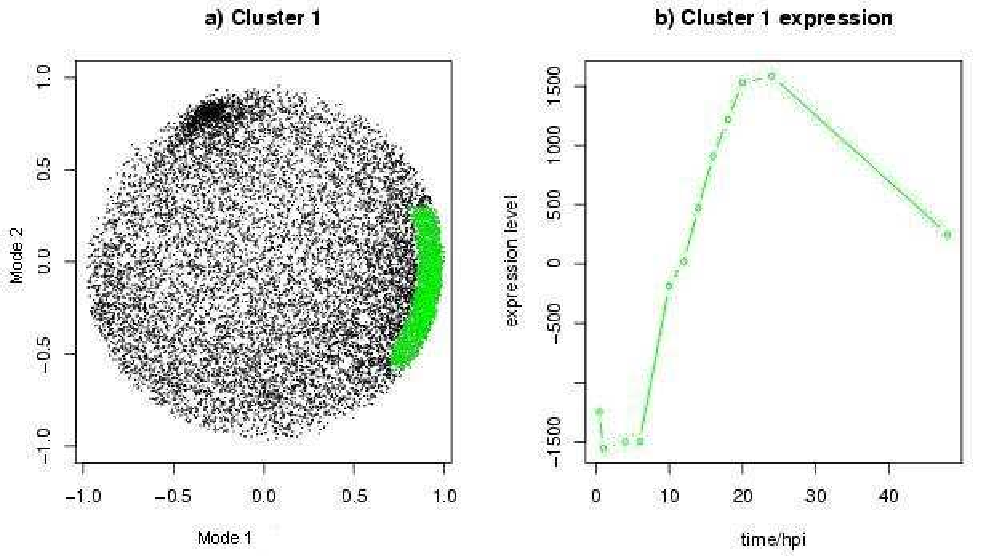


Figure 3.4: Identification of 1747 genes in cluster 1 (a) and their transcriptional response pattern (b). The 1747 genes that were correlated with mode 1, that we termed cluster 1, were selected by drawing a boundary around the region of increased density (green area in a). The transcriptional response of the genes in this cluster shows a steady increase in expression starting at 6 hpi until 24 hpi. From 24 to 48 hpi, the expression level decreases but remains above the expression values at the start of the experiment.

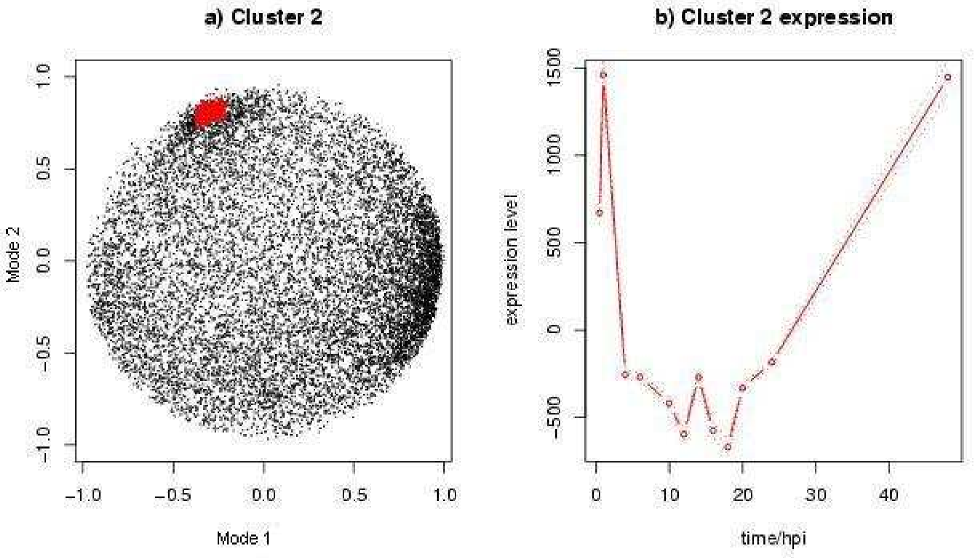


Figure 3.5: Identification of the 462 genes in cluster 2 (a) and their transcriptional response pattern (b). The 462 genes comprising this cluster were identified by drawing a boundary around the region of increased density (red area in a). The expression profile of the genes in cluster 2 shows a decrease until about 12-16 hours post infection, followed by an increase beginning at about 18 hours post infection.

host cell response to HCMV infection. These processes included signal transduction, immune system regulation, apoptosis, cell cycle regulation, oncogenesis, cell adhesion and transcription. These categories were obtained from the Gene Ontology Consortium's biological process ontology [Ashburner et al., 2000, The Gene Ontology Consortium, 2004]. Details of the annotation results can be found in [Challacombe et al., 2004]. Table 3.1 shows the distribution of the genes in the 2 clusters into the different functional categories. The annnotations of the 1747 genes in the first cluster showed 82 genes involved in immune system regulation, 73 genes involved in apoptosis, 27 genes involved in cell adhesion, 277 genes involved in transcription regulation, 155 genes involved in oncogenesis and cell cycle regulation, and 128 genes involved in signal transduction. Of the 462 genes in cluster 2, a search of the annotated gene list by biological process revealed 40 genes involved in immune system regulation, 17 genes involved in apoptosis, 20 genes involved in cell adhesion, 45 genes involved in transcription regulation, 20 genes involved in oncogenesis and cell cycle regulation, and 61 genes involved in signal transduction. Some differences between the two clusters can be seen by comparing the proportion of genes in each category to the total number of biologically relevant genes in each cluster. Comparing these numbers between cluster 1 and cluster 2 revealed a noticeably greater percentage of genes in cluster 1 in the categories of transcription (37.3%) and oncogenesis/cell cycle regulation (20.9%) than in cluster 2 (22.2% and 9.9%). Cluster

Table 3.1: Proportion of Genes in each functional category for both clusters.

Category	Cluster 1[4]	Cluster 2[5]
Immune system	82 (11.1%)	40 (19.7%)
Apoptosis	73 (9.8%)	17 (8.3%)
Cell adhesion	27 (3.6%)	20 (9.9%)
Transcription	277 (37.3%)	45 (22.2%)
Oncogenesis/Cell cycle	155 (20.9%)	20 (9.9%)
Signal transduction	128 (17.3%)	61 (30.0%)

2 contained a higher percentage of genes involved in signal transduction (30.0%), immune system regulation (19.7%), and cell adhesion (9.9%) compared to cluster 1 (17.3%, 11.1%, and 3.6%).

3.3 Summary: A new method of analysis leads to different insights

The previous study by Browne et al. [Browne et al., 2001] using Affymetrix software to analyze gene chip data found that the levels of 1425 cellular mRNAs changed by three-fold or greater in at least two consecutive time points during HCMV infection. The classes of genes affected included genes involved in immune system regulation, particularly interferon-responsive genes, genes involved in cell cycle regulation and oncogenesis, and genes whose protein products promote or inhibit apoptosis. Our dChip and SVD analysis of the same expression data resulted in two separate clusters of co-expressed genes responding differently to HCMV infection. The original analysis used GeneChip to preprocess and normalize the data and obtain expression values for the probe sets. In the analysis presented here, we used dChip to preprocess and normalize the data and obtain expression values for the probe sets. We found that dChip's multiplicative model for calculation of expression values led to lower residuals and less dependence of the residuals on the magnitude of the expression values. We then used SVD to analyze the data obtained with dChip. The SVD analysis produced two significant modes with different expression responses. These two modes captured over 75% of the variance in the data. A correlation plot of the gene expression vectors with these two modes produced two statistically significant higher density regions (clusters) of co-expressed genes that were highly correlated with mode 1 and mode 2, respectively. 26% of the genes selected by Browne et al.'s fold change filtering were present in the first cluster but only 1% were present in cluster 2. The transcriptional response pattern of cluster 2 was found to be very different from that of cluster 1. Cluster 2 genes showed a transient expression, first decreasing and then increasing again. This suggests that cluster 2 genes might be affected by the immune evasion strategies of the virus. Our results indicate that the choice of analysis methodology for gene expression data is important. While one method may work well for detecting one type of pattern in the data, it may miss another pattern altogether.

Chapter 4

New SVD based Algorithms for Gene Expression Analysis[1]

Two new algorithms for time series gene expression analysis are presented in this chapter. To motivate the algorithms, first a discussion of some clustering results of gene expression time series data is given.

4.1 Clustering of Gene Expression Time Series Data

Here we discuss the clustering results of three different studies and time series gene expression data sets [Spellman et al., 1998, Chu et al., 1998, Iyer et al., 1999]. The group of [Holter et al., 2000] used SVD to visualize the clustering results of these three studies. The data sets were a yeast cell-cycle data set [Spellman et al., 1998], a data set obtained during the yeast sporulation process [Chu et al., 1998] and a data set obtained from serum treated human fibroblast cells [Iyer et al., 1999]. The clusters of the study by Spellman et al. were obtained by first identifying all gene expression vectors that indicate cyclical cell-cycle regulation. Spellman et al. calculate for each gene a "cell-cycle score" that is supposed to capture the likelihood that a gene is cell-cycle regulated. The score is composed of the maximum correlation coefficient of the respective gene expression vector with a sine pattern of cell-cycle periodicity (the maximum is calculated over varying phase shifts), and the maximum correlation of the gene's expression vector with the expression vectors of 104 previously known cell-cycle regulated genes. Spellman et al. identified a threshold for this score at which 90% of the 104 previously known cell-cycle regulated genes had a score larger than the threshold. An additional 700 genes of the ~6000 assayed had a score above this threshold. The resulting 800 genes were then grouped into 5 clusters based on the time

[1] Part of this work was presented at the RECOMB 2003 conference [Rechtsteiner et al., 2003].

point at which their expression peaked. . The expression peaks of these 5 clusters of genes were correlated with temporal phases of the cell-cycle and labeled by these: M/G1, G1, S, G2 and M[2].

Chu et al. [Chu et al., 1998] clustered gene expression vectors for a yeast sporulation data set with 7 time points[3]. They first identified the 1100 genes, out of the 6200 yeast genes assayed on the microarrays, that showed the largest fold change, at any of the time points. 481 out of these 1100 genes showed induction during the sporulation process and the other ~620 genes showed repression. 50 genes were previously studied and known to be involved at different stages of the sporulation process. All of these 50 genes were induced genes, no genes that were repressed during sporulation had been identified previously and studied in detail[4]. Chu et al. clustered these 50 known genes into 7 groups depending on where during the sporulation process their expression peaked. The different groups were 'Rapid, transient induction (metabolic)', 'Early (I) induction', 'Early (II) induction, 'Early-middle induction', 'Middle induction', 'Mid-late induction' and 'Late induction'. Each of the newly determined 481 induced genes were then grouped into the 7 clusters based on their largest correlation with the average expression profiles of the clusters. The distribution of the induced genes into the 7 groups were 52, 62, 47, 95, 158, 61, and 6 respectively. More than half of the induced genes had not been functionally annotated previously, i.e. nothing about their function was known. Through clustering these genes with known genes, Chu et al. were able to make hypotheses about the unknown genes' functions.

Iyer et al. [Iyer et al., 1999] in their study of serum treated human fibroblast cells used cDNA microarrays with probes for ~8600 human genes. 12 time points were assayed between 15 minutes and 24 hours after serum addition to the cell culture. From the 8600 assayed genes the 517 genes with most significant change in expression were identified. The criterion for significance was that the genes either had to have a fold-change of at least 2.2 at two time points over a baseline measurement made before serum addition, or a standard deviation of the $log_2(ratio)$ over the 12 time points of at least 0.7. The 517 genes were clustered into 7 co-expression clusters with a hierarchical clustering algorithm as described in [Eisen et al., 1998].

Figure 4.1 shows the eigengene profiles found for the three data sets. The first two SVD modes captured 62% of the variance in the yeast cell-cycle data, 72% of the variance in the yeast sporulation data and 69% of the variance in the human fibroblast data. Holter et al. projected

[2] The 4 phases of the mitotic cell-cycle are G1: growth and preparation of the chromosomes for replication, S: synthesis of DNA, G2: preparation for M: mitosis, the phase in which the cell and nucleus divide.

[3] Yeast sporulation is the process of spore development, it can be induced by external signals, such as absence of nitrogen.

[4] Chu et al. speculate that the previous, non-microarray studies that had focused on the expression of individual genes and that identified these 50 genes, were biased towards finding genes that are induced. This bias could be due to some assumptions made by the experimenters, for example that sporulation is a process that requires activation of certain genes to start cellular processes and pathways related to sporulation. This illustrates how the inference (vs. hypothesis) driven and data-rich microarray technology, which allows for monitoring of many genes, can produce new insights that are difficult to make in a data-poor and hypothesis driven approach, partly due to incomplete knowledge.

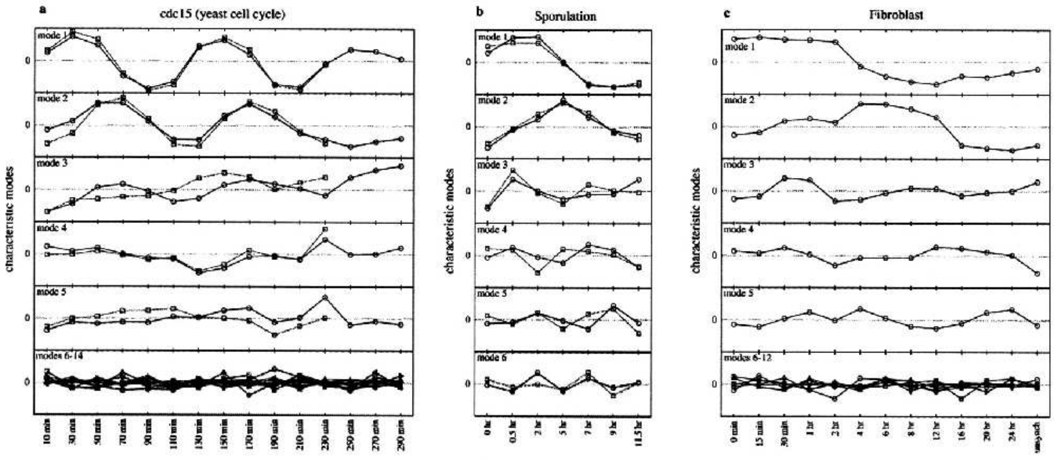


Figure 4.1: The eigengene profiles scaled by their respective singular values identified by SVD for the three time series gene expression data sets: a) yeast cell-cycle, b) yeast sporulation and c) human fibroblast cells. The eigengene profiles are ordered, the top one being the first eigengene, i.e. capturing most of the variance in the data. It is striking that the patterns of the respective first two eigengenes are rather simple, exhibiting periodic patterns (for the cell-cycle data a)), or patterns that are slowly changing with few maxima or minima (figure adapted from [Holter et al., 2000].)

the co-expression clusters of the three previous studies into the two-dimensional subspaces of the respective first two SVD eigengenes.

Holter et al. visualized the respective gene clusters by projecting their genes into the two dimensional subspace of the respective first two SVD modes (see section 2.3.1 for a discussion of scatter plot visualization of the SVD). Figure 4.2 illustrates these projections. Several observations can be made from these projections:

1. Similarly to our clusters in the herpes data, the projections in Figure 4.2 illustrate that the genes in the co-expression clusters group close to the perimeter of the two-dimensional subspace identified with SVD. This indicates that the respective first two SVD modes explain most of the expression change of the genes identified as significantly expressed in the different studies.

2. The different colors in Figures 4.2 indicate the different clusters of co-expressed genes. Genes from the same co-expression cluster do group together in the SVD subspace, indicating that proximity in the overall expression space is preserved in the respective SVD subspaces.

3. The time ordering of the expression peaks of the co-expression clusters[5] is captured and

[5] With expression profile of a cluster we usually refer to the average expression profile of the genes in that cluster.

reflected in the ordering of the clusters around the perimeter of the SVD subspaces. For example, in Figure 4.2 the clusters are ordered according to the cell-cycle phase in which the clusters' genes peak. The clock-wise ordering of the clusters in the two dimensional SVD subspace follows the temporal ordering of the phases of the cell-cycle. The cluster that peaks in M/G1 follows the cluster that peaks in G1 then the cluster that peaks in S, G2 and again M [6]. Similar temporal ordering can be observed for the sporulation and human fibroblast data sets[7]. SVD does seem to pick with the first two modes the subspace of expression where most of the expression change occurs. The subspace which captures most of the variance in the data turns out to be the subspace also capturing the dynamic change of expression during the respective biological processes.

4. The projection of the previously identified co-expression clusters allow another significant observation: although the clusters group around the perimeter of the space and a grouping (or clustering) of the genes is visible, in many cases the clusters are not very tight. At the same time, many adjacent clusters merge at their boundaries. This suggests that these time series gene expression data might not be partitioned into distinct and discrete co-expressed groups as easily as has been assumed by the application of clustering algorithms.

The latter observation is also supported by results obtained by four different clustering studies that used different algorithms to cluster the yeast cell-cycle data of [Cho et al., 1998]. All four methods, manual/visual clustering [Cho et al., 1998], Self-Organizing Maps [Tamayo et al., 1999], K-means [Tavazoie et al., 1999] and a clustering algorithm based on simulated annealing [Lukashin and Fuchs, 2001], suggest different partitionings of that same data set. The original study of [Cho et al., 1998] clustered the genes into 5 clusters, [Lukashin and Fuchs, 2001] clustered the genes into 20 clusters and the studies of [Tamayo et al., 1999, Tavazoie et al., 1999] both suggested 30 clusters, but different ones. Inspection of the cluster profiles and the expression profiles of their genes suggest that many of the smaller clusters in the studies of Lukashin et al., Tamayo et al. and Tavazoie et al. are grouped together in the larger clusters of Cho et al. Further, many clusters clearly overlap and show very similar expression profiles. Most clustering algorithms assume and search for discrete and distinct partitions of the data. If such a partitioning is not present, or the wrong number of clusters is chosen, the algorithms can impose an artificial structure on the data. The outcome then depends greatly on the specific algorithm, the parameters that were

[6] Such temporal ordering captured in a SVD subspace can also be effectively demonstrated by projecting the assays $\mathbf{a}_j$, the columns of the gene expression matrix X. Visualization of assays is also discussed in section 2.3.1.

[7] A few clusters don't fit into the temporal order as perfectly. An example is the 'Early-middle' cluster in the sporulation data, whose genes project more towards the center of the subspace and at a polar angle close to the 'Middle' genes instead onto the perimeter and at a polar angle between the 'Early II' and the 'Middle' clusters. Inspection of the average expression profiles of the different clusters in [Chu et al., 1998] suggest that the 'Early-Middle' and 'Middle' clusters have very similar expression profiles and might better be grouped together.

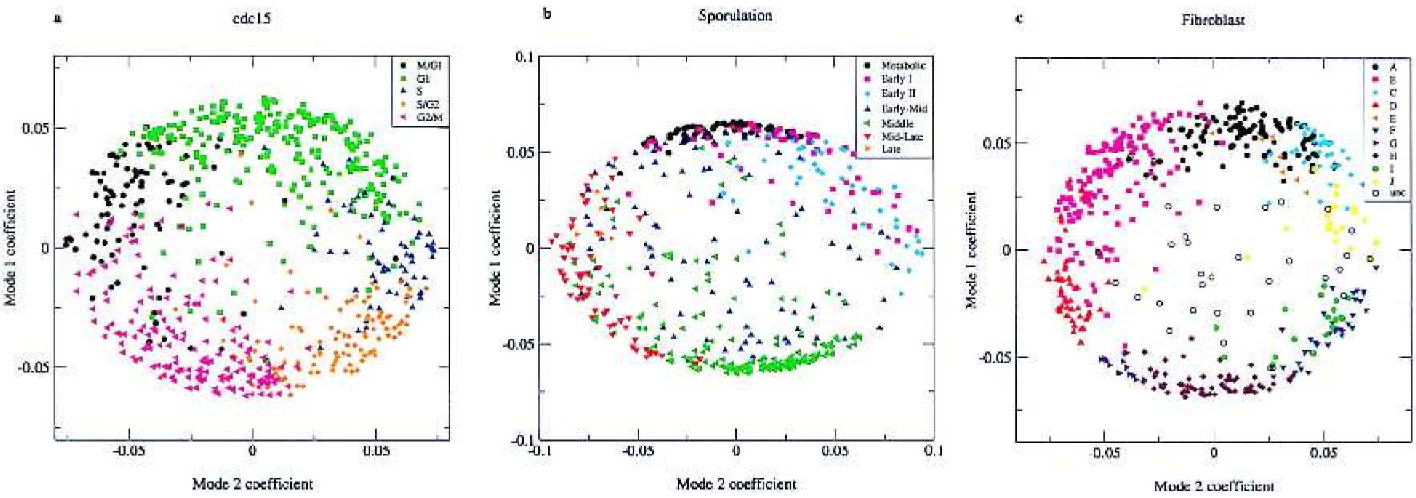


Figure 4.2: Projection of clusters obtained with different clustering techniques for different gene expression data sets. Fig a) for yeast cell-cycle data [Spellman et al., 1998], b) the yeast sporulation process [Chu et al., 1998] and c) for serum treated human fibroblast cells [Iyer et al., 1999]. The different colors indicate the different clusters identified in the different data sets[9] (figure adapted from Holter et al. [Holter et al., 2000].)

chosen, and any other initial conditions that need to be set. This illustrates the need for methods that allow for visualization of expression data and clustering results as well as other methods that can help to validate such results, for example the identification of functional coherence of genes from co-expression clusters (discussed later in chapter 5). SVD does allow for visualization and interpretation of expression data in low-dimensional subspaces and we present here two new algorithms for gene expression clustering based on projections of genes' expression vectors into subspaces spanned by two SVD modes[10]. Clustering of genes in such subspaces will allow for easy visualization of the obtained clustering results. The first algorithm identifies genes that are highly and significantly correlated with the two-dimensional expression subspace. It defines a circular boundary between genes that are likely to be significantly expressed in that expression subspace and genes that are not. The second algorithm groups the significantly expressed genes identified by the first algorithm based on their similarity in expression in the respective two-dimensional subspace.

[10] These two algorithms can be applied to the projection of genes into any two-dimensional subspace, they don't need to be subspaces identified by SVD.

4.2 Clustering of Expression Data in 2-dimensional SVD Subspaces

4.2.1 SVD allows for less aggressive "noise filtering"

All previously discussed clustering studies applied relative aggressive filtering criteria, removing between 75% and 90% of the genes assayed. Such filtering is supposed to remove the expression vectors that are noisy and non-significant. A reason for such aggressive filtering are problems of clustering algorithms with large amounts of noisy data. Noisy gene expression vectors can obscure the partitioning of the data. In section 2.3.2 we discussed the power of SVD to detect significant patterns in data even when the data is noisy. This robustness of SVD to noise allows our algorithm to work successfully with less aggressive filtering. In the applications presented here only about 50% of the genes were filtered out. It is shown in the yeast cell-cycle analysis below how known cell-cycle regulated genes that were identified in our study had been removed by aggressive filtering in the analysis of [Cho et al., 1998] and could therefore not be identified in that analysis.

4.2.2 Auto-Correlation Filter[11] for removing Noisy Genes

Our analysis approach differs from typical clustering approaches also in the filtering algorithm we developed and apply. The studies discussed above filtered genes by a fold-change approach or with a variance filter. The former requires gene expression vectors to have one (or more) expression values at least a certain factor, i.e. fold change, above the baseline expression value of the respective gene. For the baseline often the expression before the start of the experiment (i.e., t=0) or the average of the respective expression vector over the experiment is chosen. Here we applied a filter based on the autocorrelation of the expression vectors. Only genes with highest autocorrelation of their expression vectors are retained in the data set. The auto-correlation will be highest for genes that exhibit steady and relatively smooth changes in expression. This new filtering approach is motivated by the observation that in time series gene expression data, the main patterns of expression are relatively smooth and "simple". This is illustrated in the eigengene patterns displayed in Figures 4.1 and 3.2. Besides the periodic patterns of the cell-cycle data, most genes exhibit relative smooth and monotonic activation or repression with at most one reversal, i.e. maximum or minimum in expression, during these experiments. The one-step auto-correlation will be large for gene expression vectors exhibiting such patterns but will be small for genes that vary fast, e.g. expression vectors that are very noisy. A potential problem with the fold-change

[11] Also referred to as "Serial Correlation Test" in the statistics literature [Kanji, 1993].

filter is that genes that exhibit only one large peak at one time-point in their expression profile will not be filtered out, even though such expression peaks might be due to experimental artifacts. The auto-correlation filter, however, is likely to filter such genes out, as it gives more weight to smooth changing expression patterns than to the magnitude of independent gene expression values[12].

It should be noted that this filter is only applicable if the sampling rate with respect to time (or some other varying variable, for example some chemical concentration) is high enough for the process under observance, otherwise the time points are not expected to be correlated and no smooth expression patterns can be expected. If the sampling rate is too low, the auto-correlation filter we apply here is not appropriate.

Algorithm 1 Serial correlation test for filtering genes with noisy expression profiles.

1. Calculate the one-step auto-correlation for gene expression vector $\mathbf{g_i}$ which is a vector of length n (time points):

$$s_i = \frac{n}{n-1}\left\{\frac{\sum_{j=1}^{n-1}\left(x_{ij}-\bar{\mathbf{x}}_i\right)\left(x_{ij+1}-\bar{\mathbf{x}}_i\right)}{\sum_{j=1}^{n}\left(x_{ij}-\bar{\mathbf{x}}_i\right)^2}\right\} \tag{4.1}$$

   If the gene expression vectors have been standardized to mean zero, this simplifies to:

$$s_i = \frac{n}{n-1}\left\{\frac{\sum_{j=1}^{n} x_{ij}x_{ij+1}}{\sum_{j=1}^{n} x_{ij}^2}\right\} \tag{4.2}$$

2. If s_i is smaller than some critical value s_c, remove gene i from the data set. Alternatively, remove a certain fraction of genes with lowest s_i.

In the gene expression analysis work here the auto-correlation filter outlined in Algorithm 1 was applied. For each gene i, the auto correlation coefficient s_i is calculated. Large s_i indicate that the sequential expression measurements of gene i are correlated, and thus unlikely to be of random nature. Either a threshold s_c can be determined below which genes with $s_i \leq s_c$ will be removed, or a predetermined fraction of genes with lowest s_i can be removed.

4.2.3 Boundary Identification in Two-dimensional Spaces (BITS)

It was shown that the significantly expressed genes in several time series expression data sets are highly correlated with the first two SVD eigengenes, i.e. they project towards the perimeter of the subspace spanned by the first two eigengenes. The algorithm presented here defines a circular

[12] Due to the outlined potential problems, the fold-change filter has sometimes been modified. For example [Browne et al., 2001] required two expression values at consecutive time points to be above a fold-change threshold.

boundary in the two-dimensional subspace that separates the significantly expressed genes that are highly correlated with the two eigengenes from the genes that project towards the center of the subspace. Similarly to the synthetic data in Figure 4.3, or the “real” expression data in Figure 3.3, gene expression vectors which are not highly correlated with the two eigengenes, and therefore project towards the center of the correlation plot, are expected to be more uniformly distributed in the two-dimensional subspace. Because they are weakly correlated with the eigengenes, their location in the two-dimensional subspace is mostly influenced by noise. The significantly expressed genes that are highly correlated with the subspace and whose expression is due to some underlying biological process are expected to be less uniformly distributed in the subspace. This difference in the distribution of the genes in the subspace is used to define a circular boundary separating the significantly expressed genes from the non-significantly expressed genes.

The algorithm assesses the uniformity of the distribution of the genes by first calculating the density estimate of the distribution of the polar angles of the genes in the subspace. Consider again Fig. 4.3 or Fig. 3.3. Close to the center of the correlation plot the distribution of the polar angles of the genes will be close to uniform, i.e. the density function of the genes over the interval $[0,2\pi]$ will be close to one-dimensional uniform density with value $1/2\pi$. Further away from the origin and closer to the perimeter, the distribution of the genes will become less uniform, as the significantly expressed genes involved in different biological processes are expected to be differently regulated and expressed. There will in most cases be a less sharp boundary in real gene expression data than there is found in Fig. 4.3. But variation in the density of genes highly correlated with the first two SVD modes can definitely be observed in real data, as for example in Figures 4.2 and 3.3. Note further that gene expression vectors projecting towards the center, though not having significant expression in that specific subspace, might have significant expression in another subspace orthogonal to the one being observed. Although we have shown previously that the first two eigengenes seem most times to suffice to capture the significant change in expression, the algorithm might be applied iteratively to different subspaces of SVD eigengenes.

Mathematical Details of the BITS algorithm

See Algorithm 2 for a detailed listing of the steps of the algorithm. Let us denote the orthonormal vectors spanning the two-dimensional space by $\mathbf{v}_{k_1}$ and $\mathbf{v}_{k_2}$. In the application here $\mathbf{v}_{k_1}$ and $\mathbf{v}_{k_2}$ will be two SVD eigengenes of the expression data matrix X. The correlation vectors $\mathbf{c}_{k_1}$ and $\mathbf{c}_{k_2}$ contain the correlation coefficients r_{ik_1} and r_{ik_2} (see equation 2.13) of the gene expression vectors $\mathbf{g}_i$ with the eigengenes $\mathbf{v}_{k_1}$ and $\mathbf{v}_{k_2}$respectively. $c_{k_1}^{(i)}$ denotes the ith element of $\mathbf{c}_{k_1}$ and $c_{k_1}^{(i)} = r_{ik_1}$. If $\mathbf{v}_{k_1}$ and $\mathbf{v}_{k_2}$ are SVD eigengenes and the gene expression vectors $\mathbf{g}_i$ have been centered to have zero mean and are normalized to unit length, then the SVD provides the correlation coefficients in the respective eigenassays scaled by the singular values, i.e. $\mathbf{c}_{k_j} = \mathbf{u}_{k_j} s_{k_j}$ (see Eqn. 2.14).

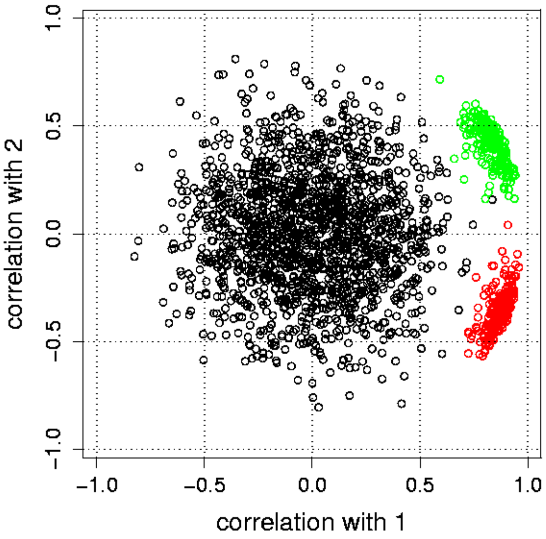


Figure 4.3: Correlation plot of synthetic data set introduced in section 2.3.1. Figure reproduced from Fig. 2.5. The bottom right cluster (red) is composed of the sine wave genes, and the top right cluster (green) is composed of the exponential decay genes. The cluster of genes around the origin corresponds to the noise-only genes.

The correlation plot of the expression vectors $\mathbf{g}_i$ onto the 2-dimensional subspace of $\mathbf{v}_{k_1}$ and $\mathbf{v}_{k_2}$is obtained by plotting the coefficients in $\mathbf{c}_{k_1}$ against the coefficients in $\mathbf{c}_{k_2}$. Note that $c_{k_1}^{(i)2} + c_{k_2}^{(i)2} \leq 1$, $i = 1,..,m$. Therefore the correlation plot will be a disk of radius one.

An initial radius r_0 (see step 2 in Alg. 2) is chosen and the density $\hat{f}_r$ of the polar angles of the gene expression vectors $\mathbf{g}_i$ inside the circle with radius r_0 in the correlation plot is calculated. The only requirement on the starting value r_0 is that the number of gene expression vectors inside the circle with radius r_0 allow for a stable density estimation[13]. $e(r)$, a measure of the distance of $\hat{f}_r$ from the uniform density $1/2\pi$, is calculated (see step 3c in Alg. 2). According to our earlier reasoning, for small r, $\hat{f}_r$ is expected to be close to the uniform density and $e(r)$ therefore close to zero. r is iteratively increased and $\hat{f}_r$ and $e(r)$ are recalculated. As the radius r increases the distribution of genes closer to the perimeter of the space becomes more structured and $\hat{f}_r$ will deviate more from the uniform distribution. A boundary is defined by determining the radius $\tilde{r}$ at which the rate of change of $e(r)$ is largest.

It should be noted that no parameter value needs to be specified for this algorithm. The algorithm will identify the boundary $\tilde{r}$ in a data driven way. For example, $\tilde{r}$ will depend on the level of noise in the data. More noise in the data will make the region around the origin of uniformly

[13] A value of $r_o = 0.4$ has been found to be appropriate for the data sets the algorithm was tested on.

Algorithm 2 Estimate the separating boundary $\tilde{r}$ by finding the greatest rate of change in the density function of the polar angles for the genes inside the circle of radius $\tilde{r}$.

```
1. Select two orthonormal directions $\mathbf{v}_{k_1}$ and $\mathbf{v}_{k_2}$ (in our case SVD eigengenes) and compute the correlation vectors $\mathbf{c}_{k_1}$ and $\mathbf{c}_{k_2}$.
2. Chose an initial value for the radius: $r_0$. Then assign: $r \Leftarrow r_0$.
3. while $r < 1$ do
    (a) Find the set of genes inside the circle with radius $r$: $I_r \Leftarrow \{i : {c_{k_1}^{(i)}}^2 + {c_{k_2}^{(i)}}^2 \leq r^2, i = 1, ...n\}$.
    (b) Compute the one-dimensional density, $\hat{f}_r$, of the polar angles of the genes in $I_r$.
    (c) Compute the value of $e(r) = \text{median}_{j \in I_r}\{|\hat{f}_r(\phi_j) - \frac{1}{2\pi}|\}$, which is a measure of the deviation of the density $\hat{f}_r$ from the uniform density $\frac{1}{2\pi}$, over the support of the polar angles $[-\pi, \pi]$.
    (d) Assign a new value to $r$ for the next iteration: $r \Leftarrow r + h$.
4. end while
5. Find the boundary $\tilde{r}$ that maximizes the rate of change of $e(r)$, i.e. that maximizes $de/dr$.
```

distributed genes larger, the algorithm will therefore tend to determine a larger boundary $\tilde{r}$ than for lower levels of noise.

4.2.4 Polar Angle Density Clustering (PAD Clustering)

A second algorithm was developed to define groups of similarly expressed genes around high density regions in the band of genes identified by the BITS algorithm. The motivation for this algorithm is the same as for clustering algorithms in general: co-expressed genes might be functionally related. Once the boundary $\tilde{r}$ in the 2-dimensional expression subspace has been identified by the BITS algorithm (Alg. 2), the distribution of the genes in the ring with radius $\tilde{r} \leq r \leq 1$ can be inspected visually. If the distribution of genes suggest that there are regions with significantly higher density of genes, these regions can be clustered into groups of similarly expressed genes by the Polar Angle Density Clustering algorithm outlined in Alg. 3.

First, all the local maxima in the density function of the distribution of the polar angles that are greater than the uniform density $1/2\pi$ are identified. The algorithm then forms partitions by grouping together all gene expression vectors with polar angles around a peak and with a density value larger than the uniform density. For each peak in the density a group of similarly expressed genes is obtained. Note that this partitioning algorithm is again data driven. The number of groups

Algorithm 3 Polar Angle Density (PAD) Clustering: Identify clusters of co-regulated genes.

```
1. Apply Algorithm 2 and find $\tilde{r}$.
2. Identify the set of genes outside $\tilde{r}$: $I_{\tilde{r}} \Leftarrow \{i : c_{k_1}^{(i)2} + c_{k_2}^{(i)2} \geq \tilde{r}^2, i = 1, \ldots, m\}$.
3. Compute the density, $\hat{f}_{\tilde{r}}$, of the polar angles for the genes in $I_{\tilde{r}}$.
4. Identify the genes with maxima in their one dimensional density and with values above the uniform density: $S_c \Leftarrow \{j : \hat{f}_{\tilde{r}}(\phi_{\mathbf{j}})$ is a local maximum of $\hat{f}_{\tilde{r}}$ and $\hat{f}_{\tilde{r}}(\phi_{\mathbf{j}}) > 1/2\pi$, $j \in I_{\tilde{r}}$, $\phi_j \in [-\pi, \pi]\}$.
5. Let $s_{(j)}$ be the ordered values of $S_c$ ($\phi$-values at the peaks).
6. $nb_c \Leftarrow \mathrm{card}(S_c)$ (the number of detected peaks)
7. $h_1 \Leftarrow 1/2\pi$
8. for $j = 1$ to $nb_c$ do
    (a) $lwr \Leftarrow \min\{m : m < s_{(j)}, \hat{f}(m) > h_j$ and $\hat{f}(m) < \hat{f}(m+)\}$(left boundary)
    (b) $upr \Leftarrow \max\{m : m > s_{(j)}, \hat{f}(m) > h_j$ and $\hat{f}(m) > \hat{f}(m+)\}$(right boundary)
    (c) $cluster_j \Leftarrow \{k : lwr < \theta_k < upr$, where $\theta_k$ is the polar angle for gene $k$ $\}$ (genes in the cluster)
9. end for
```

of similarly expressed genes identified by the algorithm is data dependent and does not need to be specified, in contrast to some clustering algorithms like K-means and SOMs.

4.2.5 Extension to higher Dimensions

The algorithm outlined above works in a 2-dimensional subspace. To have a similar 3-dimensional implementation of the algorithm, the genes inside a sphere would have to be projected onto the 2-dimensional surface of that sphere and the distribution of the genes for different sphere sizes would have to be compared (similarly as outlined in Alg. 2 for one-dimensional projections for varying sizes of circles). Studies of such a 3-dimensional version of the algorithm revealed that too few data points, i.e. gene expression vectors, are available to populate the space sufficiently to have stable estimates of the density of genes projected onto the 2-dimensional spheres. The algorithm can, however, be applied iteratively. For example, if 3 significant eigengenes have been identified for a gene expression data set, different 2-dimensional projections spanned by two of the three eigengenes can be explored for structure in the projection.

4.3 Application of BITS and PAD Algorithms to Cell-Cycle Data

The above illustrated method and its algorithms were applied to the yeast cell-cycle data of [Cho et al., 1998] [14]. The main goal of the study was to identify the cell-cycle regulated genes in yeast. [Cho et al., 1998] used the Affy chip technology to measure 6200 yeast genes at 17 time points taken at ten-minute intervals, spanning two cell-cycles. The data was first transformed by taking the logarithm, and each gene expression vector was standardized to have zero mean and unit standard deviation. The auto-correlation filter outlined in Alg. 1 was applied and the 50% genes with the lowest auto-correlation coefficient were removed. SVD was applied to this filtered gene expression data set. The singular value spectrum and the first three eigengenes are shown in Fig. 4.4. The first three SVD modes account for 31%, 19% and 14% of the total variance. The first eigengene shows a pattern of steady decrease or, for genes which are anti-correlated to this pattern, increase in expression. It has been observed that the first eigengene is often associated with some large trend affecting all or many genes in the data set [Alter et al., 2000] [15]. The monotonic increase or decrease in expression of eigengene 1 could be due to the effects of the artificial synchronization of many yeast cells for the experiment. Cells are arrested in a certain phase of the cell-cycle and all are released from this artificial arrest at time point zero. It is likely eigengene 1 captures the large-scale ’relaxation’ of the cellular system back to a ’steady state’[16]. Eigengenes 2 and 3 show

[14] The data was already briefly introduced in section 2.3.1.

[15] If the data is not standardized to mean zero, the first mode usually represents the average expression of the genes.

[16] Similar observations and suggestions have been given in the data set analyzed by [Alter et al., 2000].

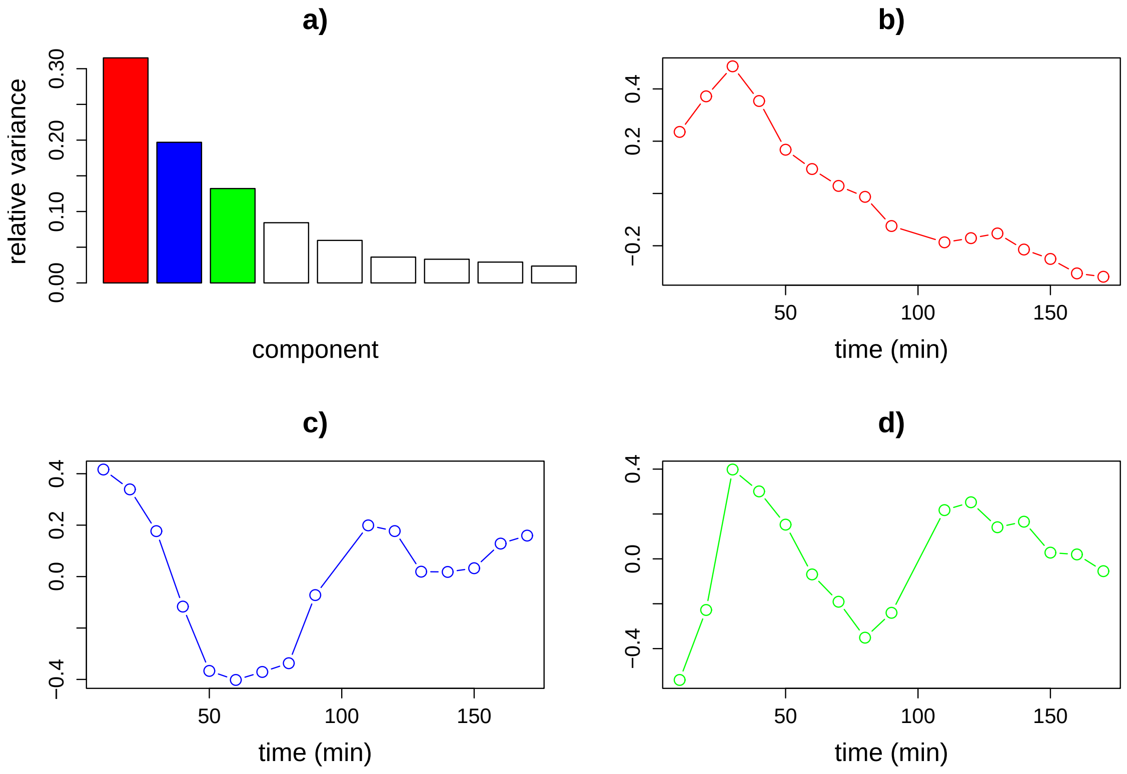


Figure 4.4: SVD of the Cho et al. [Cho et al., 1998] yeast cell-cycle gene expression data. Plots of relative variance (a); and the first (b), second (c) and third (d) eigengenes are shown (same Figure as Fig. 2.2).

the cyclic patterns we expect to find in cell-cycle expression data. The periodicity of the patterns is close to the length of the cell-cycle and their phase difference is close to $\pi/2$. Cell-cycle regulation is associated with changes in the expression that is periodic with the cell-cycle. Identification of cell-cycle regulated genes is therefore typically associated with identifying genes with expression patterns that show a periodicity with the cell-cycle. Other studies [Spellman et al., 1998, Cho et al., 1998, Tavazoie et al., 1999] have used the same association of periodic expression patterns with cell-cycle regulation. To identify the most significant periodic gene expression vectors, the BITS 2 and PAD clustering algorithm 3 were therefore applied to the expression data set with eigengenes 2 and 3 spanning the two-dimensional expression subspace. The algorithms are applied to identify genes that are cell-cycle regulated.

Note that eigengene 2 and 3 are not perfect sine patterns. For example, their amplitudes decay over time. Such features are likely to be real properties of the data, the individual gene expression vectors. The decay in amplitude, for example, is probably due to the loss of cell-cycle synchronization between the cells over time. Deriving $\mathbf{v}_{k_1}$ and $\mathbf{v}_{k_2}$ from the data with, for example, SVD

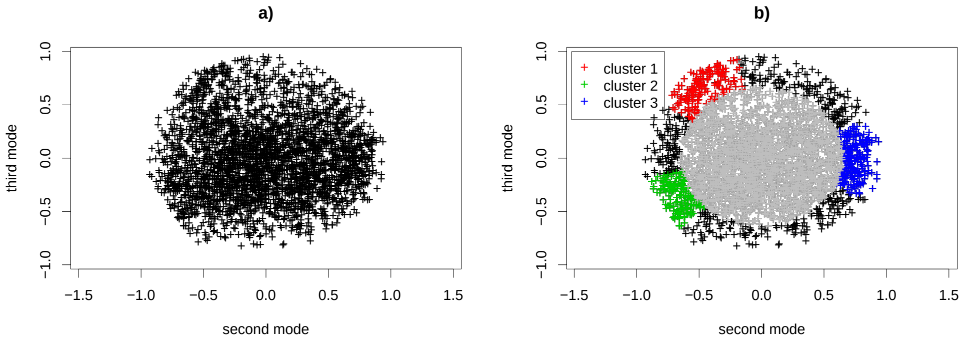


Figure 4.5: a) Correlation plot of the yeast cell-cycle data with eigengenes 2 and 3. b) Correlation plot with the 3 high-density regions detected by algorithm 3.

instead of using some idealized pattern like a sine and cosine therefore promises in many cases to be a better approach.

4.3.1 Application of the BITS algorithm

Figure 4.5 a) shows the correlation plot for the cell-cycle data with eigengenes 2 and 3 as $\mathbf{v}_{k_1}$ and $\mathbf{v}_{k_2}$. Algorithm 2 estimated the circular boundary at $\tilde{r} = 0.67$. Algorithm 3 was used to identify clusters of similarly expressed genes outside of $\tilde{r}$. Figure 4.6 shows two plots of the density estimates of the polar angle distributions for genes with radius $r \leq 0.4$, the starting value r_0, and for genes in the outer ring with radius $r \geq 0.65$. The change of the distributions from a close to uniform distribution to a much less uniform distribution is apparent[17].

Three higher-density regions in the distribution of the polar angles outside of $\tilde{r}$ are visible. Figure 4.5 b) shows the correlation plot of the gene with the clusters detected by Algorithm 3. Figure 4.7 shows the expression patterns of all the genes in the three detected clusters. Sub-figure d) shows the average expression pattern for each cluster. Each cluster of genes shows a clearly periodic expression pattern, but with different phases. The number of genes outside the circle with radius 0.67, i.e. potentially cell-cycle regulated, is 895. The largest cluster (Figure 4.7 c)) contains 206 genes, the second largest (Figure 4.7 b)) contains 164, and the third one (Figure 4.7 a)) contains 152 genes.

[17] The number of genes is 1027 and 985 respectively, and the bandwidth for the kernel density estimator was set at 0.25. The difference in the densities is therefore not due to to unequal sample size or the bandwidth parameter.

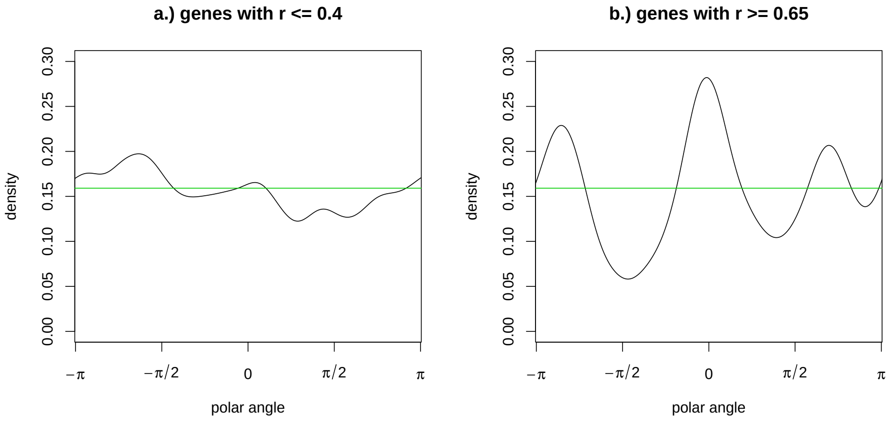


Figure 4.6: Density estimates of polar angle distributions for genes with radius a.) $r \leq 0.4$ and b.) $r \geq 0.65$. The change in the distribution from a fairly uniform to a much less uniform distribution is apparent.

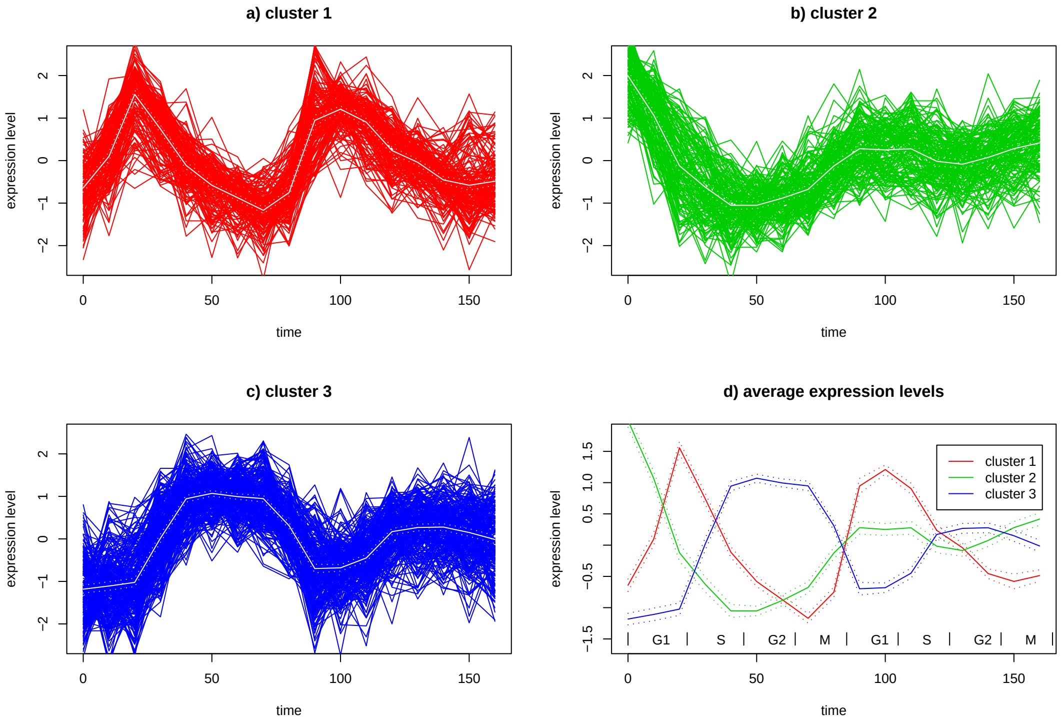


Figure 4.7: Three different clusters of similarly expressed genes identified by Algorithm 3 after application of Algorithm 2. Figures a), b) and c) show all the expression patterns of the genes in the clusters. Figure d) shows the average expression patterns for the three clusters.

4.3.2 Statistical Test of Significance of Results

Estimation of false positive rate.
One method to assess the quality and reliability of gene expression results is based on statistical tests. For example, one can generate synthetic, random data with similar statistical properties as the real data and assess the likelihood of finding the same results by chance. Such a statistical technique can be used to estimate a false positive rate associated with the obtained results. Here we want to estimate the number of genes that would obtain a high correlation with the profiles $\mathbf{v}_{k_1}$ and $\mathbf{v}_{k_2}$ by chance only, due to noise in the data. One method to estimate the false positive rate is to generate 'control data' by performing random permutations among the elements of each gene expression vector $\mathbf{g_i}$ from the original data matrix X [Yeung et al., 2001b]. By permuting the elements of the gene expression vectors $\mathbf{g}_i$, the distribution of the expression values in the new data matrix is maintained, the dependence between the time points is broken, however. The 'random genes' are simulated without making any assumption about the distribution of the noise in the data. If the simulated random genes are projected into the space of original eigengenes 2 and 3 (the eigengenes used to obtain the above results), the average number of random genes that fall outside the circle of radius $\tilde{r} = 0.67$ is 42. The estimate of the false positive rate of this method is then 42/895=0.046, which amounts to less than 5%.

4.3.3 Biological Significance of Results

The overall biological significance of each cluster in Figure 4.7 was investigated by first associating the clusters with phases of the cell cycle. The association was made by plotting the time-dependent average expression profile of each cluster in the context of cell-cycle phase information. Figure 4.7 d) shows that the expression of genes in cluster 1 (red) peaks in the late G1 phase of the cell-cycle, genes in cluster 2 (green) peak in the M/G1 phase, and genes in cluster 3 (blue) peak in the G2 phase. The biological significance of individual genes within each cluster was explored making use of the KEGG database [Kanehisa Lab., 2005].

Two observations are reported. The first observation was found by searching the KEGG text annotations of all of the yeast genes in the expression data set for instances of the text string 'transcr' (to capture anything associated with 'transcription') and further hand-selecting 184 genes that were annotated as being involved in transcription. Of these 184 genes, 32 were found within the set of genes that our analysis predicted as cell-cycle regulated. Of these 32, only one is found within cluster 1 (peaking in late G1): SWI4, which is a known cell-cycle regulator (and also reported by Cho et al. as a cell-cycle regulated gene). It therefore seems that transcription-related genes are underrepresented in cluster 1.

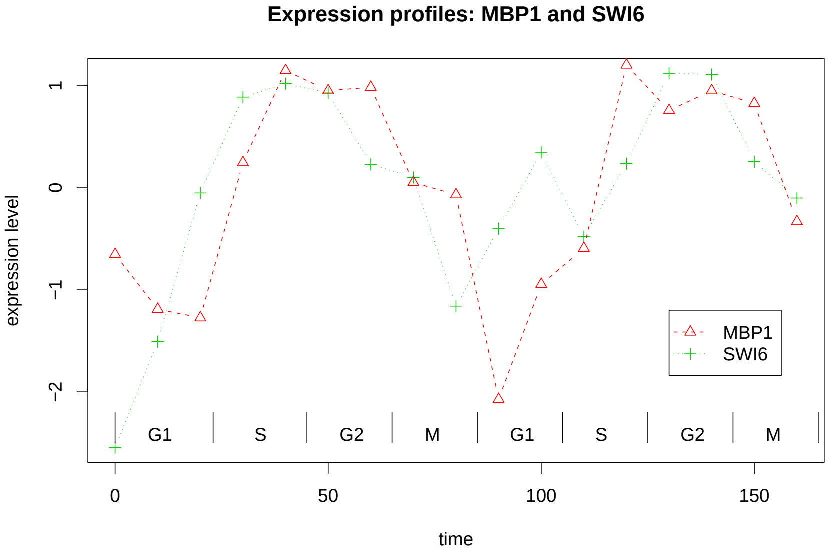


Figure 4.8: Expression profiles of the two genes coding for SWI6 and MBP1 respectively. Both show clearly periodic expression profiles.

An important observation we are able to report is the inclusion of genes encoding SWI6 and MBP1 among our predicted cell-cycle regulated genes. These genes were not included among cell-cycle genes identified by Cho et al., despite being known cell-cycle regulators and both genes showing clearly cyclical expression profiles with a periodicity of the cell-cycle 4.8. This observation leads to the question why these obvious candidates for cell-cycle regulated genes were missed by Cho et al., even more so considering that Cho et al. selected their cell-cycle candidate genes by visual inspection. See the next section of a discussion of this.

4.3.4 Discussion of Results

Comparison of the results to the original study by Cho et al.

Cho et al. [Cho et al., 1998] reported 416 cell-cycle regulated genes of which 231 agree with our 895 genes. The differences in the number of cell-cycle regulated genes identified by both studies can be explained by several differences in the studies. Cho et al. filtered the gene expression data set with a fold-change approach and were left with only 1300 genes for further analysis. These 1300 genes were then inspected visually for periodic patterns by Cho et al. Among the 895 genes that were detected as potentially cell-cycle regulated by our method about 600 were removed by the fold change criteria used by Cho et al. Visual inspection of these 600 genes showed that most of them clearly had cyclic expression patterns. Among the 600 genes removed by Cho et al.'s fold-change approach are the genes coding for SWI6 and MBP1, both known to be cell-cycle regulated.

Figure 4.8 shows the expression profiles of the two genes, they clearly exhibit cyclic patterns with a periodicity of the cell-cycle. These results suggest that the fold-change approach of Cho et al. filtered out most of the 600 genes with cell-cycle related periodic expression patterns because their amplitude was too low to pass the fold-change threshold. This illustrates one problem with the fold-change filter. Gene expression patterns clearly significant for the specific experiment at hand can be missed if the amplitude of the pattern is too low or the baseline expression of the gene is high in general. On the other hand, because the fold-change filter, in general, treats the different time-points as independent and does not consider the overall pattern of the gene expression time series, it can pass genes with a single peak in expression. Such single peaks might be artifacts if they are the only significant expression change during the whole experiment, however.

The ($\simeq$ 175) genes that Cho et al. declared cell-cycle regulated but were not identified as such in our study were also visually inspected. Many of these genes showed one significant peak in their expression response. The experiment was conducted over two cell-cycles, however, and cell-cycle regulated genes would be expected to peak at 2 time points separated by approximately one cell-cycle period. How these genes were declared cell-cycle regulated by Cho et al., although they exhibit only one peak in two periods, has not been resolved.

Biological findings.

Our analysis of yeast cell-cycle data has led to two biologically significant findings. The first finding is based on the observation that transcription-related genes are relatively underrepresented in cluster 1 (red). As cluster 1 is associated with late G1 phase of the cell cycle (Figure 4.5 b.), the expression of transcription-related genes in the Cho et al. data set seems relatively repressed among cell-cycle regulated genes in late G1 phase. The SWI4 mRNA transcript (part of the SWI4/SWI6 complex, which modulates Cln1, Cln2, Cln6 and Swe1 is the only one that was found to be relatively abundant in late G1, perhaps to poise the cell for response upon upregulation of SWI6. This finding leads to a hypothesis that cell-cycle genes that code for transcription factors are relatively silent in late G1 phase. One way to rationalize such a tendency is by noting that late G1 phase corresponds to “stop” in budding yeast, a point where progression through the cell cycle can be arrested if the proper environmental signals are not received. Here we suggest the possibility that late G1 phase in part prepares the cell for the possibility of cell-cycle arrest by decreasing regulation of cell-cycle related gene expression, a hypothesis that can be tested by further experiments. The second finding gives evidence for cell-cycle regulation of SWI6 and MBP1 genes. The expression profiles of SWI6 and MBP1 are shown in Figure 4.8, and clearly show periodicity characteristic of cell-cycle genes. The SWI6 and MBP1 protein products are the molecular constituents of the SBF complex, a known cell-cycle regulator that modulates expression of Cln1, Clb6, Clb5, Gin5 and Swe1. The SWI6 and MBP1 genes were not identified by Cho et al. as cell-cycle genes.

Chapter 5

Automated Mining and Analysis of Functional Information for Gene Expression Data[1]

5.1 Introduction

Gene expression data analysis has mostly focused on mining the numerical expression data for significant expression patterns and gene co-expression clusters. But ultimately the biological meaning of any numerical analysis results needs to be identified. For example, the biological function of genes from a co-expression cluster in the context of the experiment needs to be found. Sources of such information are annotations of genes and proteins in databases and functional information about them contained in the literature. Traditionally, biology experts have been mining these sources of information manually. When experiments are designed to test a single hypothesis, and few genes or proteins are involved, such an approach is manageable. However, with the advent of high-throughput techniques like microarrays in Functional Genomics, where hundreds of genes can make up a co-expression cluster, the development of automated algorithms that can assist in knowledge discovery will become increasingly important.

Here such a method for automated knowledge discovery for groups of genes (or proteins) from literature is presented. In short, we present a method that takes genes that cluster in expression space and finds if these genes also cluster in a *functional space,* derived from the literature. Where genes project in expression space is independent from where they project in literature space[2].

[1] Some of the work outlined here was presented at the RECOMB 2004 conference and the Rocky 1 Bioinformatics workshop [Rechtsteiner and Rocha, 2004a, Rechtsteiner and Rocha, 2004b], a publication is in preparation.

[2] Assuming that literature about expression experiments does not (yet) dominate the literature to an extent where they actually are not independent anymore.

Finding genes that cluster in expression space and also cluster in the functional literature space will therefore 1.) support (or validate) the found expression clusters as significant and 2.) provide functional information about the respective clusters.

The method presented here accepts a group of genes, e.g. genes that are co-expressed in an expression experiment, and identifies what we term *functional themes* with which these genes are associated in the literature. Knowledge contained in the literature is represented by the Medical Subject Heading (MeSH) terms [National Library of Medicine, 2004], an indexing vocabulary of the biomedical literature database MEDLINE/PubMed[3] [National Library of Medicine, 2005]. Literature for the genes is obtained from the curated protein sequence database SwissProt/UniProt [SIB/EBI, 2004]. The algorithm used to mine the literature information for relevant knowledge about the groups of genes is derived from the *vector space model* of Information Retrieval (IR) [Baeza-Yates et al., 1999, Rijsbergen, 1979].

In the original vector space model of IR, documents are represented as vectors in a so-called *keyword* or *term space,* typically the terms contained in the whole set of documents or some vocabulary that is used to index the documents (see also Fig. 5.1). Similarly to representing documents in a term space, we represent genes in MeSH term space. Documents that are relevant for a gene are obtained, then the MeSH terms that index these documents in the MEDLINE database are retrieved. The documents that are obtained are publications referenced in the expert curated protein sequence database SwissProt. Obtaining the literature from a curated database like SwisProt insures that the quality of the publications and their relevance for the respective genes is high. If genes have similar biological functions, the respective documents hopefully discuss these functions, which will be reflected in the MeSH terms indexing these documents. The vectors of functionally related genes in MeSH term space is therefore expected to be similar. We explore the gene-MeSH term space for significant groups of genes that are functionally related, and the MeSH terms associated with these genes expressing their functional themes, with Singular Value Decomposition (SVD).

5.2 Data and Methods

5.2.1 The MeSH Vocabulary

MeSH is the National Library of Medicine's controlled vocabulary thesaurus. It consists of sets of terms and naming descriptors in a hierarchical structure that permits searching at various levels of specificity. At the most general level of the hierarchical structure are very broad headings such as *Anatomy* or *Enzymes*. More specific headings are found at more narrow levels of the eleven-level

[3] PubMed is the WWW gateway to MEDLINE.

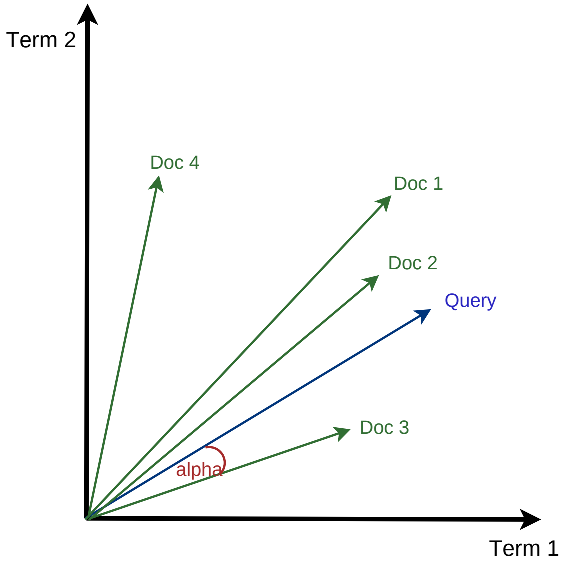
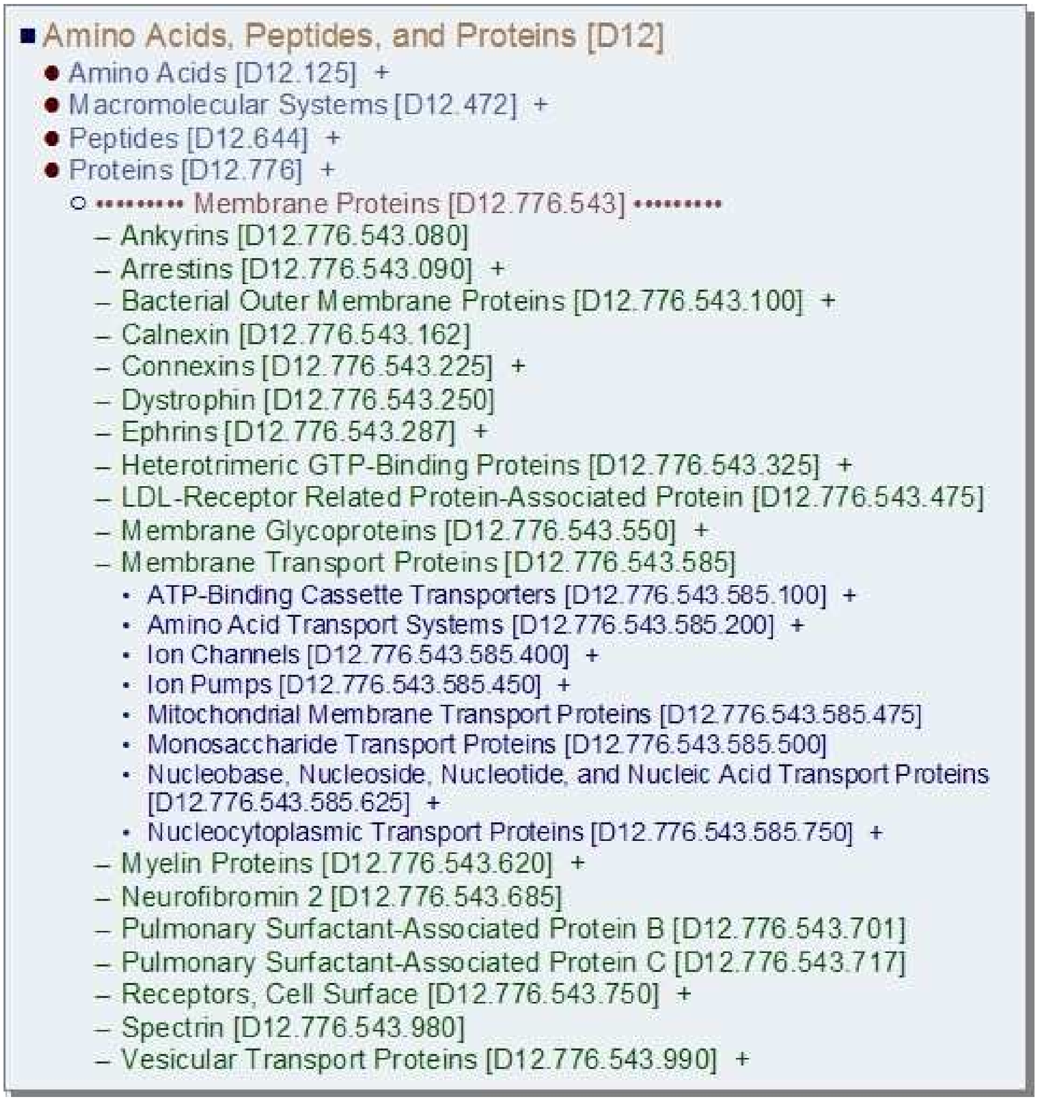


Figure 5.1: The document-term vector space model in Information Retrieval. Documents and the query terms are represented as vectors in term space. As a similarity measure between document vectors and document and query vectors the cosine of the angle between the vectors is typically chosen. Documents can then be retrieved and ranked by their decreasing cosine similarity with the query vector.

hierarchy, such as *Ankle* and *Lactose Synthase*. Part of the MeSH subtree under *Amino Acids, Peptides, and Proteins [D12]* is shown in Figure 5.2. The MeSH thesaurus is used by NLM for indexing articles from 4,600 of the world's leading biomedical journals for the MEDLINE/PubMed database. Each bibliographic reference is associated with a set of MeSH terms, an average of about a dozen, that describe the content of the item. Similarly, search queries sent to PubMed [National Library of Medicine, 2005] use the MeSH vocabulary to find publications on a desired topic.

MeSH contains over 22,000 main headings, or terms, and over 100,000 synonyms to these (also referred to as "entry terms"). The MeSH vocabulary is continually revised and updated. Subject specialists are responsible for areas of the health sciences in which they have knowledge and expertise. In addition to receiving suggestions from indexers and others, these experts collect new terms as they appear in the scientific literature or in emerging areas of research, they define these terms within the context of existing vocabulary and they recommend their addition to MeSH.

MeSH has been used in some other studies as the vocabulary of choice for biomedical knowledge discovery [Masys et al., 2001, Jenssen et al., 2001]. [Masys et al., 2001] chose two small groups of 25 genes each that were differently expressed in two types of leukemia [Golub et al., 1999]. They obtained gene identifiers for the 50 genes and searched PubMed and retrieved 70 publications related to the 50 genes. They showed that the MeSH terms occurring more frequently



- ■ Amino Acids, Peptides, and Proteins [D12]
  - ● Amino Acids [D12.125] +
  - ● Macromolecular Systems [D12.472] +
  - ● Peptides [D12.644] +
  - ● Proteins [D12.776] +
    - ○ ••••••••• Membrane Proteins [D12.776.543] •••••••••
      - – Ankyrins [D12.776.543.080]
      - – Arrestins [D12.776.543.090] +
      - – Bacterial Outer Membrane Proteins [D12.776.543.100] +
      - – Calnexin [D12.776.543.162]
      - – Connexins [D12.776.543.225] +
      - – Dystrophin [D12.776.543.250]
      - – Ephrins [D12.776.543.287] +
      - – Heterotrimeric GTP-Binding Proteins [D12.776.543.325] +
      - – LDL-Receptor Related Protein-Associated Protein [D12.776.543.475]
      - – Membrane Glycoproteins [D12.776.543.550] +
      - – Membrane Transport Proteins [D12.776.543.585]
        - ATP-Binding Cassette Transporters [D12.776.543.585.100] +
        - Amino Acid Transport Systems [D12.776.543.585.200] +
        - Ion Channels [D12.776.543.585.400] +
        - Ion Pumps [D12.776.543.585.450] +
        - Mitochondrial Membrane Transport Proteins [D12.776.543.585.475]
        - Monosaccharide Transport Proteins [D12.776.543.585.500]
        - Nucleobase, Nucleoside, Nucleotide, and Nucleic Acid Transport Proteins [D12.776.543.585.625] +
        - Nucleocytoplasmic Transport Proteins [D12.776.543.585.750] +
      - – Myelin Proteins [D12.776.543.620] +
      - – Neurofibromin 2 [D12.776.543.685]
      - – Pulmonary Surfactant-Associated Protein B [D12.776.543.701]
      - – Pulmonary Surfactant-Associated Protein C [D12.776.543.717]
      - – Receptors, Cell Surface [D12.776.543.750] +
      - – Spectrin [D12.776.543.980]
      - – Vesicular Transport Proteins [D12.776.543.990] +

Figure 5.2: Part of a subtree of the MeSH concept hierarchy. Every '+' sign indicates there are more entries below that entry. Only the subtree *Membrane Proteins* is shown under *Proteins*. Note that going down the hierarchy, the terms become more specific.

with the publications of the two sets of genes were meaningful in the biomedical context of the two types of leukemia. Our approach is different from [Masys et al., 2001] in several ways. We apply our method to far larger groups of genes (hundreds to thousands) and with far larger sets of literature (> 19,000 publications in the study presented here). Further, our algorithm will identify functional themes that can be represented by many MeSH terms and from different regions in the hierarchical tree (e.g. a theme containing "Enzymes" and associated "Diseases"). The resulting functional themes can be used to group the genes from a co-expression cluster into different functional groups and filter out genes that were not associated with any such themes and therefore more likely to be in the co-expression cluster due to noise in the expression data. The 50 genes of [Masys et al., 2001] were selected conservatively, with high confidence that they are differently expressed in the different conditions. Our much larger gene clusters are expected to contain much more noise and identification of significant MeSH terms and functional themes associated with the clusters is more difficult[4]. [Jenssen et al., 2001] built a gene network from co-occurrences of gene identifiers in abstracts of MEDLINE. They annotated the links among pairs of genes with the MeSH terms that index the publications mentioning the respective gene pairs in MEDLINE. Chapter 6 in this work presents a large scale study that evaluates the clustering of functionally related groups of proteins in MeSH space[5].

5.2.2 The Vector Space Model and Latent Semantic Analysis

Figure 5.1 illustrates the vector space model of IR [Baeza-Yates et al., 1999] (see also [Deerwester et al., 1990, Berry et al., 1995]). Given a set of documents, the words (or terms) are extracted from the documents, or they are obtained from an indexing vocabulary that is used to index the publications (i.e. here the MeSH vocabulary is used which indexes publications in MEDLINE). If the terms are extracted from the documents directly, typically a so called *stop-list* is applied to remove frequent and general terms that are not informative about the contents of the documents. Each document can then be assigned a term vector which contains as the vector coefficients the number of times the respective term occurs in the document[6]. These so-called *term frequencies* (tf) are also referred to as the *local weight* of the respective term for the document [Dumais, 1990]. A so called *global weight* for each term is usually applied as well. This global weighting is supposed to capture the information content of the respective term in the respective body of documents.

[4] In [Masys et al., 2001] an Internet address to an online tool implementing their methodology is mentioned. I wanted to apply the tool to the data presented here and compare the results but I have never been able to access the tool and an e-mail inquiry to the authors was not answered.

[5] The study of chapter 6 partly grew out of the attempt to try to quantitatively and objectively validate MeSH as a vocabulary for biochemical knowledge discovery, a question that arose from the work in this chapter.

[6] Alternatively, the coefficients can be the log of the number of times the term occurs in the document, which reduces the weight of very frequently occurring words.

The most commonly applied global weighting is the Inverse Document Frequency (IDF) [Dumais, 1990] weighting. Given a term t_k, occurring in n_k documents, and N being the total number of documents, the IDF weighting factor for term t_k is defined as $idf_k = log(\frac{N}{n_k})$. Considering the extreme values of idf_k illustrates the effects of this weighting: idf_k is maximal for terms that only occur in one document ($idf_k = log(N)$), these terms have much predictive power. Terms that occur in all N documents, however, have no predictive power or information content about different documents and their weighting factor is $idf_k = log(\frac{N}{N}) = 0$. The document vectors in term space can then be represented in a document-term matrix, e.g. the documents as columns and the terms as rows. The coefficient of document vector d_i at term dimension t_k is then given by the matrix element

$$w_{ki} = tf_{ki} * idf_k \tag{5.1}$$

where tf_{ki} is the term frequency of term t_k in document d_i and idf_k the previously discussed IDF for term t_k. Similarly to representing documents in term space, a set of query terms can be represented as a vector in term space. A common similarity measure between document vectors (and between document and query vectors) is the cosine of the angle between the term vectors. Given two document vectors $\mathbf{d}_i$ and $\mathbf{d}_j$ in term space, the cosine similarity between the two vectors is defined by the normalized dot product:

$$cos(\mathbf{d_i}, \mathbf{d_j}) = \frac{\mathbf{d_i}\mathbf{d_j}}{|\mathbf{d_i}||\mathbf{d_j}|} \tag{5.2}$$

where $|\mathbf{d_i}|$ and $|\mathbf{d}_j|$ denote the Euclidean lengths of document vectors $\mathbf{d_i}$ and $\mathbf{d_j}$. Similarly document vectors can be compared to a query term vector. Given a query vector, documents can then be retrieved and ranked by decreasing cosine similarity with the query vector.

Here the vector space model of IR was adapted for the representation of gene vectors in MeSH term space. First, relevant literature for all the genes on the microarray chip for which the analysis was performed needed to be obtained. We obtained the publications referenced by the respective genes in the SwissProt database (see details in next section). The MeSH terms for the publications were obtained from MEDLINE. For each gene g_i and MeSH term m_k, the number of publications referenced by gene g_i and also indexed by MeSH term m_k are counted, this number represents our local weight, gene-MeSH term frequency mf^g_{ki}. Each MeSH term m_k was weighted by a global weighting factor similar to IDF: given the number of all genes for which we have literature, N^g, and the number of genes that reference publications that are indexed by MeSH term m_k, n^g_k, we define a global weighting called the Inverse Gene Frequency (IGF) for MeSH term m_k:

$$igf_k = log(\frac{N^g}{n_k^g}) \tag{5.3}$$

Similar to IDF, if a MeSH term occurs with all N^g genes, IGF for this MeSH term is zero, as this MeSH term cannot be informative about different genes. If a MeSH term occurs with only one gene, IGF is maximal, $igf_k = log(N^g)$, as this MeSH term is potentially very informative about the gene and its function. The gene vectors in MeSH term space can then be represented in a gene-MeSH term matrix. Similar to Eqn. 5.1, the coefficients of gene vector g_i (columns of matrix) in MeSH term dimension m_k (rows) is given by the matrix value

$$w_{ki}^g = mf_{ki}^g * igf_k \tag{5.4}$$

Latent Semantic Analysis of MeSH Term Space

Given a cluster of co-expressed genes, we can project these genes into MeSH term space as outlined above. We can now search for groups of genes that cluster in MeSH term space. Here we search and identify these genes and their location in MeSH term space with Singular Value Decomposition. If genes cluster in a certain location in MeSH space, we expect the variance of the gene-MeSH data in that direction to be larger and SVD will be able to detect these higher variance directions. The MeSH terms associated with these SVD modes will describe the *functional themes* the groups of genes are associated with.

SVD is frequently applied in combination with the vector space model in Information Retrieval, it is then typically referred to as Latent Semantic Analysis (LSA) or Latent Semantic Indexing (LSI) [Deerwester et al., 1990, Berry et al., 1995]. Similarly to our application to genes in MeSH term space, SVD can detect if documents cluster into different themes and subjects. SVD modes have been found to be associated with such different themes and subjects, therefore the term “Latent Semantic” (LS) space, which in our application we can correspondingly call “Latent Functional” space.

Besides identifying the dominant themes and subjects in a body of documents, it has been found that the application of SVD to the vector space model can significantly improve retrieval of documents. Using only the top SVD modes (typically a couple of hundred in a data set with thousands of documents and thousands of terms [Berry et al., 1995]), LSA leads to a reduction of the dimensionality of the document-term space. This reduction has often the beneficial effects that 1.) unimportant and “noise introducing” terms are ignored, as they are typically captured by the low variance singular vectors and 2.) that projection of the document vectors into the SVD subspace reduces the negative effects of term *synonymy* and *polysemy* on IR with the vector space

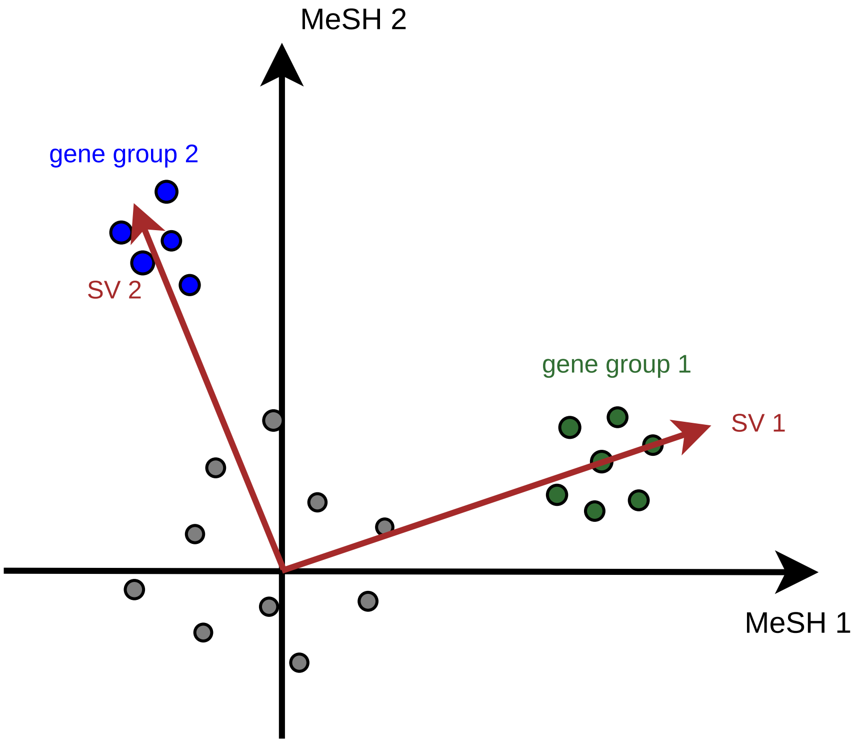


Figure 5.3: If groups of genes cluster in MeSH term space, SVD can detect the directions of these groups of genes. The illustration shows two groups of genes that cluster in MeSH term space. One singular vector (SV 1) points in the direction of group 1 and the second singular vector points in the direction of gene group 2. The grey genes in the center are not strongly associated with either MeSH term 1 or MeSH term 2.

model. Multiple terms are synonyms of each other if they have the same meaning (e.g. "car" and "automobile"). They should therefore not be treated as independent terms and ideally "collapse" to the same location in the reduced LSA space (e.g. "car" and "automobile" are expected to be close in the reduced LSA space). Polysemy refers to one term having multiple meanings (e.g. "Java" in the different contexts of programming language, coffee or as an island). The ambiguity of polysemic terms can be reduced in LSA space. If "Java" occurs in a document projecting close to documents about coffee in LSA space, it will most likely have a different meaning in that document (i.e. meaning related to coffee) than if the document projects close to documents about computers (i.e. probably refers here to the Java programming language). The meaning of the same terms in different documents can therefore be disambiguated by LSA based on where in LSA space the documents project to. In chapter 6 we predict protein sequence families for proteins based on where in MeSH term space the proteins project. We also explored if the application of SVD/LSA to the protein-MeSH term space could improve our prediction success. We found that this is not the case and that many (over one thousand) singular vectors are necessary to obtain comparable results to predicting in the original MeSH term space (~5000 MeSH terms). This seems to indicate that there is little synonymy or polysemy in the MeSH term vocabulary[7]. This might be expected from a well designed vocabulary, as ambiguous indexing terms in MEDLINE would certainly pose a problem for the database and retrieval of relevant documents. Although SVD/LSA in gene-MeSH term space will not be needed for disambiguating MeSH terms, it will still identify any directions in MeSH space in which genes cluster.

5.2.3 Obtaining the MeSH Term Frequencies mf_{ki}^{g}

The associations of genes with MeSH terms, the mesh term frequencies mf_{ki}^{g} in Eqn. 5.4, need to be obtained to perform the above outlined analysis. To obtain the MeSH terms, we first need to obtain documents that discuss the functions of the genes. One possibility is to obtain gene names and symbols from gene databases and query MEDLINE titles and abstracts for these gene identifiers. The gene names and symbols usually have high ambiguity, though. There can be multiple identifiers for the same gene (synonymy of gene identifiers) or the same identifier can refer to multiple genes or concepts (polysemy). [Jenssen et al., 2001] inferred gene networks from co-occurrences of gene identifiers in abstracts and titles of MEDLINE publications. They found that 30-40% of the inferred connections in the network were incorrect due to synonymy and polysemy of gene identifiers. Here we therefore chose a different approach and obtained what could

[7] In document retrieval applications significant improvements in document retrieval have been reported when reducing the dimensionality from originally thousands to only several hundred dimensions. Most of the improvement is due to the reduction of term synonymy in the reduced latent semantic space [Deerwester et al., 1990, Berry et al., 1995].

be called *expert literature* for the genes. We obtained the literature from the expert curated protein sequence database SwissProt/UniProt [SIB/EBI, 2004, Bairoch et al., 2005] from the European Bioinformatics Institute (EBI). Each protein entry in SwissProt contains the sequence of the protein, protein and gene names and identifiers pooled from various other databases. SwissProt also contains cross-references to to other databases, for example the gene sequence in GenBank or the Pfam protein sequence family the respective protein belongs to. Each protein entry also has references to relevant literature for the protein in MEDLINE. Because of expert curation, we can have high confidence in the relevance of the literature referenced in SwissProt.

See Figure 5.4 for the steps involved in obtaining the gene-MeSH term data. To obtain literature for the genes from an expression experiment, we needed to obtain a mapping of gene identifiers from the respective mRNA chip to SwissProt proteins. In the data set analyzed below, the mRNA chip was manufactured by Affymetrix [Affymetrix, 2005], and Affymetrix does provide a mapping of the genes on their chips to SwissProt proteins[8]. SwissProt provides us with literature references for the respective proteins (or genes). The MeSH terms for these literature references are obtained from MEDLINE.

5.3 An Application of LSA to Gene-MeSH Space

5.3.1 Three Gene Expression Clusters in Herpes Virus Infected Human Cells

The above outlined automated functional analysis was performed on three co-expression clusters from the herpes virus infected human fibroblast data set discussed in chapter 3. The expression of 12,600 genes (probe sets) was measured with Affymetrix chips (HGU95A) at 12 time-points, between 1/2 hrs and 48 hrs after infection with the herpes virus. To eliminate the genes with mostly noisy expression profiles the one-step auto-correlation filter introduced in chapter 4 was applied and half of the genes with lowest one-step autocorrelation were removed. Singular Value Decomposition was applied to identify the dominant modes of expression for the remaining genes. Figure 5.5 shows the singular value spectrum and the first two eigengene profiles, the first exhibiting a monotone increasing expression pattern and the second a transient pattern of initial decrease and then increase in expression. 80% of the variance in the expression data was captured by these first two expression modes.

Plotting the correlation of the gene expression vectors with eigengenes 1 and 2 showed strong variation in the density of genes around the perimeter of that subspace. The boundary identification (BITS) and polar angle density clustering (PAD) algorithms introduced in chapter 4 were applied

[8] Even without such a mapping, typically a gene to SwissProt protein entry mapping is not too difficult to obtain, as SwissProt contains extensive cross-references to gene databases like GenBank.

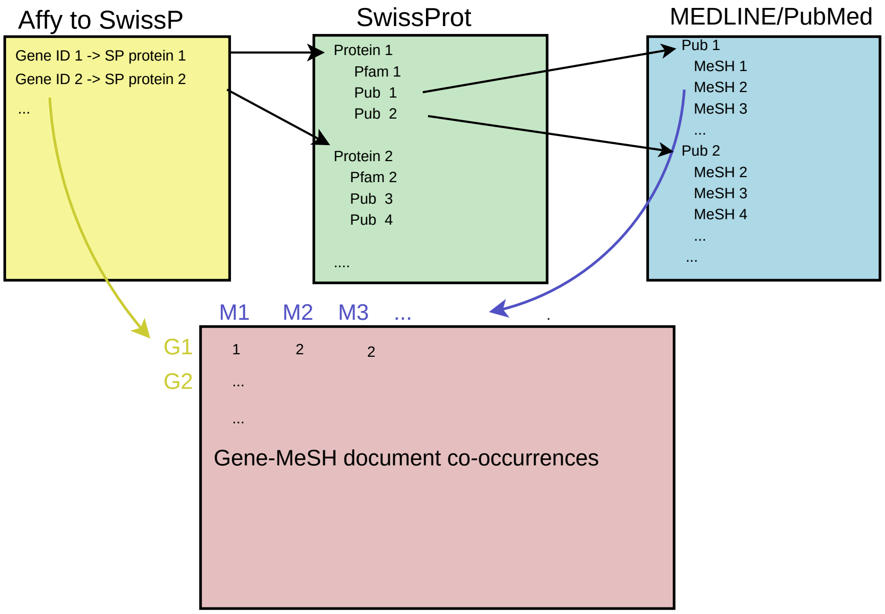


Figure 5.4: To obtain the gene-MeSH term association matrix, we obtained a mapping of Affymetrix gene IDs for the chip used in the gene expression analysis (HGU95A) to SwissProt proteins. The SwissProt database provided us with literature references for the proteins. MeSH terms for these publications were obtained from MEDLINE. The gene-MeSH association matrix then contains for each gene-MeSH term pair the number of documents referenced by the respective gene and also indexed by the respective MeSH term in MEDLINE (i.e. the MeSH term frequencies mf_{ki}^{g} from Eqn. 5.4).

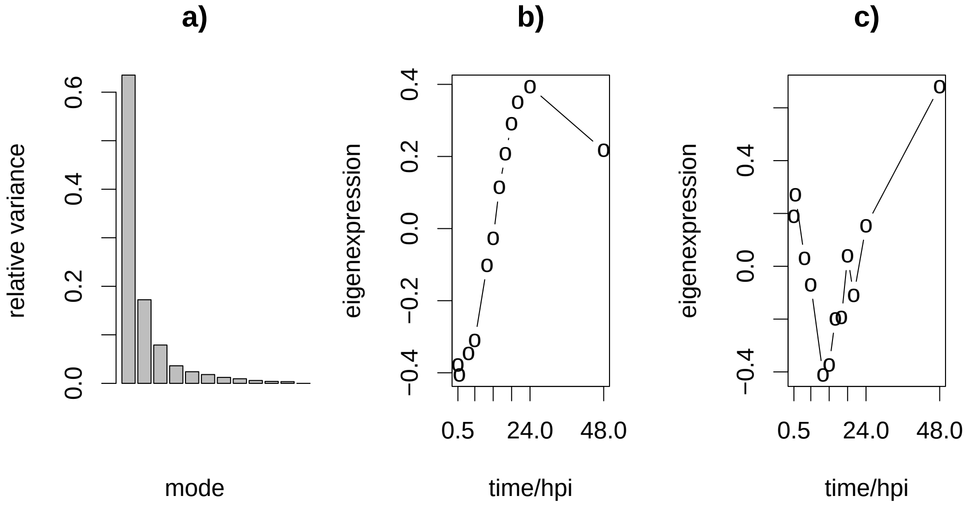


Figure 5.5: Singular Value Decomposition of the herpes infected human fibroblast data set. Shown is the singular value spectrum (the relative variance captured by the respective modes) and the first two eigengenes. Over 80% of the variance is captured by the first two expression modes.

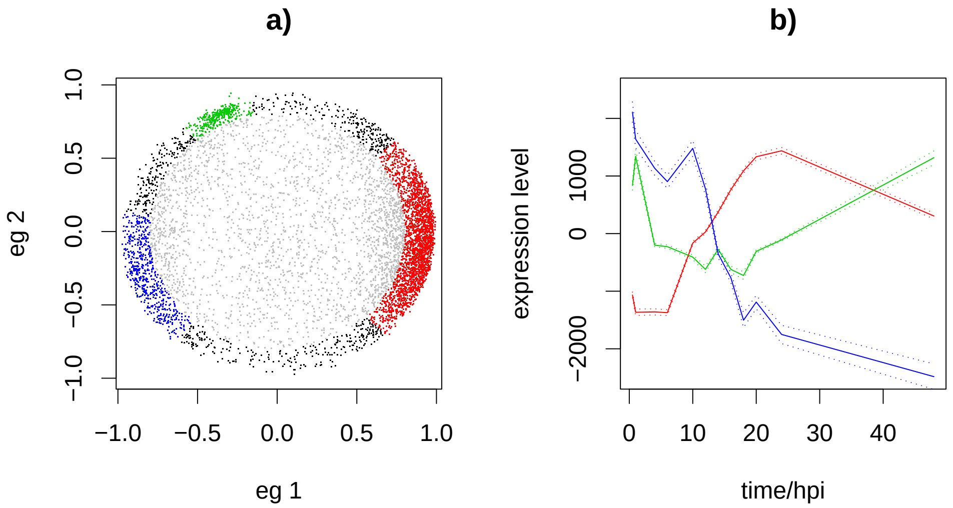


Figure 5.6: Results of the application of the BITS and PAD algorithms to gene expression vectors in the subspace of eigengenes 1 and 2. Three high density regions of genes were identified close to the perimeter of the space (shown in plot a). The average expression profiles of the genes in the respective clusters are shown in plot b).

to the gene expression vectors in the subspace of eigengenes 1 and 2. Three high density regions, or clusters, of similarly expressed genes were identified, see Figure 5.6 a). Two of the clusters (red and green) were identified in the study presented in chapter 3. An additional region with a higher density of genes (blue) was identified by the PAD algorithm. Figure 5.7 shows a bi-plot, a projection of the genes and the assays onto SVD modes 1 and 2, i.e. the projection of the genes onto the first two eigengenes and a projection of the assays onto the first two eigenassays[9]. For easier interpretation, the sign of the projection of the assays onto eigenassay 2 has been flipped. The red arrows corresponding to the assays are labeled by the time points (hrs after infection). The figure shows that the assays are time ordered in this two-dimensional projection, they are ordered clock-wise. The assays also cluster in roughly 3 groups, and these groups of assays project close to the 3 clusters of co-expressed genes. Genes that project close to a certain assay vector in the bi-plot have their expression induced at the corresponding time point (relative to the baseline to which expression is measured, here the mean expression over all time points). The blue cluster

[9] Possibly the easiest way to visualize this plot is to think of the genes plotted as points in the original 12 dimensional space spanned by the 12 assays, i.e. time points of the experiment. These time point basis vectors are illustrated by the red arrows. SVD performs a rotation of the space so that most of the variance is captured by the new basis vector system (the singular vectors, linear combinations of the original basis vectors). Figure 5.7 shows the projection (rotation) of the genes (black dots) and the original time point basis vectors (red arrows) onto the first two singular vectors capturing most of the variance. This is achieved by projecting the genes onto the eigengenes (right singular vectors) and projecting the assays onto the eigenassays (left singular vectors). (See also chapter 2.)

of genes is associated with assays corresponding to the early time point (1/2, 1, 2 and 4 hrs after infection). These genes' expression is induced at these early time points, as can be seen in the expression profiles in Figure 5.6 b). The assays at 10 and 12 hrs project close to the genes of cluster 2. Because these assays project close to eigenassay 2, and the sign of the projection with eigenassay 2 has been flipped, genes of cluster 2 have a repressed expression at these intermediate time points. Assays corresponding to medium to late time points, i.e. from 14 to 48 hrs after infection, project close to gene expression vectors of the red cluster (where the last, 48 hr time point, is somewhat outlying). These genes will have an induced expression at these medium to late time points (as can again be seen in the average expression profile for the cluster of genes in Figure 5.6 b)).

In addition to the 12 time point assays after virus infection, Browne et al. [Browne et al., 2001], who produced the study, also measured expression after a so-called "mock infection". In mock infection, the human fibroblast cells undergo the same experimental procedure as in infection with the herpes virus except that no virus is present. Such mock infections are performed to control for effects of the experimental procedure, and not the virus, on gene expression of the host cells. Two assays, 16 and 24 hrs after mock infection, were obtained as controls. The projection of both of those assays onto eigenassays 1 and 2 are shown in Figure 5.7 as green arrows, labeled m16 and m24. Both mock infected assays project close to the assays corresponding to the earliest time points of the virus infection experiment, e.g. the assays taken 1/2 and 1 hr after virus infection. Compare the mock infected assays also to the assays at 16 and 24 hrs after virus infection. The assays 16 and 24 hrs after virus infection are positively correlated with mode 1 whereas the mock infected assays, corresponding to the same time points after mock infection, are negatively correlated with mode 1. This suggests that, first, the mock infected cells are in a similar "state of expression" as the cells right after infection. It suggests further that not many changes in gene expression have occurred 1/2 and 1 hr after infection with the herpes virus. We can conclude further that the change in expression we observe in the subspace of mode 1 and 2 are solely due to the virus and not any experimental procedures during infection.

5.3.2 LSA in MeSH Term Space

For 11,348 Affymetrix IDs on the chip HGU95A mappings to SwissProt protein identifiers were obtained[10]. 8036 of the Affymetrix IDs had mappings to 6074 SwissProt protein identifiers (multiple Affymetrix probe sets can be present for the same gene/protein; the maximum number of Affymetrix probe sets for a protein were 10). The 6074 genes/proteins had 19644 publications referenced in SwissProt. MeSH terms for these publications were retrieved from MEDLINE. The

[10] Downloaded from the Affymetrix website [Affymetrix, 2005], requires (free) registration.

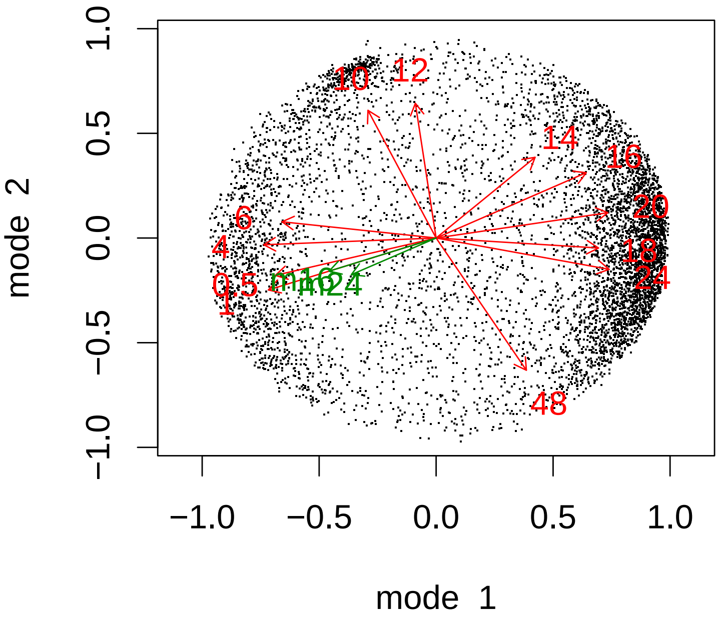


Figure 5.7: Projection of both the gene expression vectors (black dots) onto the eigengenes and the assay expression profiles onto the eigenassays (red arrows). The assays are labeled by the hour after infection they were sampled at. The assays seem to group roughly into 3 distinct groups which project close to the three clusters of genes that were identified by the PAD algorithm. Also, the assays are in temporal, clockwise order around the perimeter, suggesting that some temporal process(es) is in progress causing the observed changes in expression. Also shown are two assays 16 and 24 hrs after a “mock infection” (green arrows). These two assays project close to the earliest assays, 1/2 and 1 hr, after virus infection. This indicates that the observed expression change in the 12 assays after virus infection are due to the virus and not any experimental effects of the infection.

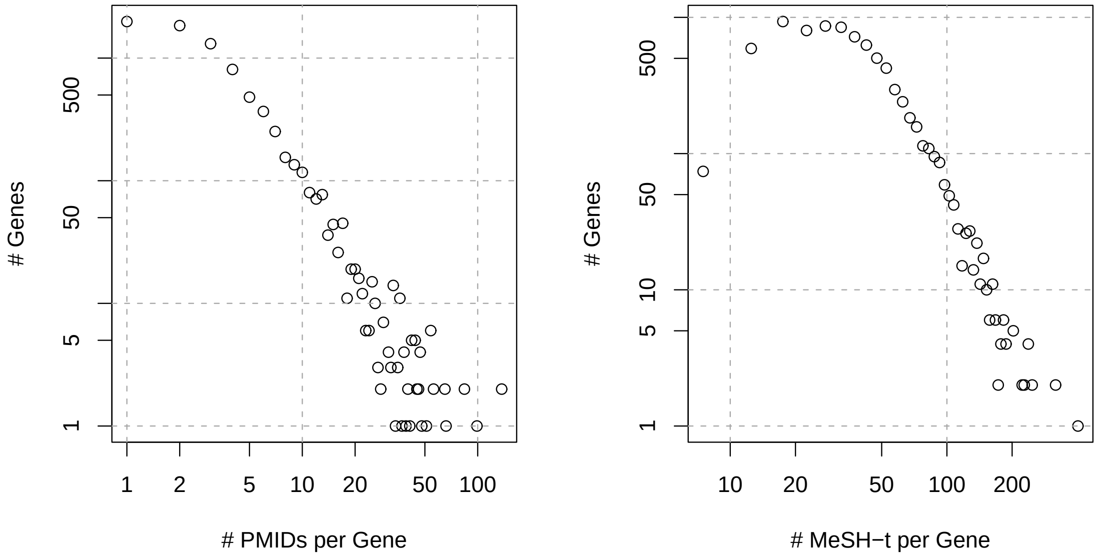


Figure 5.8: The distribution of number of publications (PMIDs) and MeSH terms per gene (or protein) in our data set. Many genes have only 1 or 2 publications referenced. The mean number of publications per gene is 4.3, the median is 3. The mean number of distinct MeSH terms associated with a gene is 42, the median is 35.

19644 publications were found to be indexed by 6568 distinct MeSH terms. The 346 MeSH terms that indexed publications of more than 100 different proteins were removed, as they are most likely too general to be informative about a gene's (or protein's) function. The 1928 MeSH terms that indexed publications of only one gene were removed as well, as such very unique terms often add mostly noise to the data and do not link any proteins/genes in MeSH term space. It has been found that typically the terms occurring with 'medium frequency' are the most important for successful information retrieval with the vector space model [Deerwester et al., 1990].

We obtained a 6074 by 4294 gene-MeSH association matrix, with a matrix element for a gene-MeSH pair indicating how many publications were found referenced by the respective gene/protein in SwissProt and also indexed by the respective MeSH term in MEDLINE (the mesh term frequency mf_{ki}^{g} in Equation 5.4).

For the 4294 MeSH terms the inverse gene frequency (IGF) was calculated (see Eqn. 5.3) and applied as global weighting factors to the MeSH term frequency matrix, see Eqn. 5.4. 1528 Affymetrix probe sets from cluster 1 mapped to 1343 distinct genes in SwissProt. 210 probe sets of cluster 2 mapped to 205 distinct genes in SwissProt and 383 probe sets from cluster 3 mapped to 339 distinct genes.

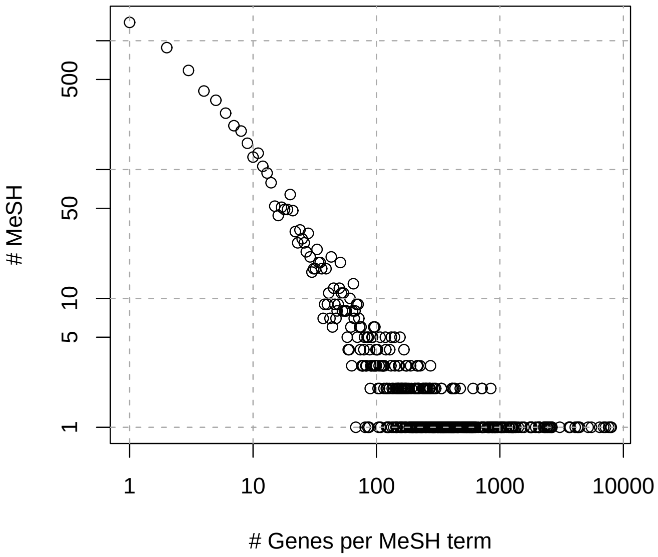


Figure 5.9: The number of genes a MeSH term occurs with (linked through the publications referenced in SwissProt). Some MeSH terms occur with almost all genes but many occur with only one or few genes. The 5 most frequently occurring MeSH terms are Human, Molecular Sequence Data, Amino Acid Sequence, Support - Non-US Govt and Base Sequence. Obviously the very frequent MeSH terms are not informative about specific genes.

1. Dihydrotestosterone receptor; Androgen receptor)
2. Amyloid beta A4 protein precursor; ABPP; Alzheimer's disease amyloid protein
3. Hemoglobin beta chain
4. A-28; Aw-68; HLA class I histocompatibility antigen, A-68 alpha chain precursor
5. Transthyretin precursor; Prealbumin; TBPA; TTR; ATTR
6. Collagen alpha 1(I) chain precursor
7. Cystic fibrosis transmembrane conductance regulator; CFTR; cAMP-dependent chloride channel
8. Serine-protein kinase ATM
9. vWF; Von Willebrand factor precursor
10. Thyroid stimulating hormone receptor; TSH-R

Table 5.1: Genes with most publication references. Synonym names and symbols are separated by semicolons.

Iterative Algorithm for MeSH Outlier Removal

Before applying SVD to the resulting gene-MeSH association matrices for the three gene co-expression clusters, random sets of genes with the same sizes as the clusters were generated to identify and remove "outlier MeSH terms". Outliers here refers to MeSH terms that alone can determine a SVD mode of a random set of genes and therefore can bias the analysis. 15 random sets of genes, 5 sets with the same size as cluster 1 (1343 randomly selected genes), 5 sets with the same size as cluster 2 (205 randomly selected genes), and 5 sets with the same size as cluster 3 (339 randomly selected genes) were created. SVD was performed on the resulting 15 gene-MeSH association matrices and the outliers among the MeSH terms were identified. Outlier identification and removal is rather typical in data analysis with PCA/SVD [Jolliffe, 1986] and often performed by visual inspection of the data. Here we developed an algorithm that performs outlier detection and removal in an automated way. The algorithm identifies for a random gene-MeSH data set the MeSH terms that are most correlated (largest absolute value of the coefficient) with each of the first 10 SVD modes (of that respective gene-MeSH data set). If the coefficient of a MeSH term for one of the 10 SVD modes is larger than 0.7, the MeSH term is marked as outlier and for removal before the next iteration of the algorithm[11]. All outlier MeSH terms for all 15 random sets and for the first 10 singular vectors were marked and then removed. The algorithm was applied to the resulting data set and the next set of MeSH term outliers were determined. After 10 iterations, when fewer than 5 MeSH terms were identified per iteration, the threshold for the singular vector coefficients was lowered to 0.6 $(0.6^6 = 0.36)$. After 14 iterations, again less than 5 MeSH terms were identified as outliers in the 15 random sets. It was also found that the singular value spectrum for 15 different, independent random data sets, obtained as controls, stopped to vary significantly between iterations. The algorithm to remove MeSH term outliers was therefore halted after 14 iterations and the removal of 158 MeSH term outliers. Most of the removed MeSH term outliers are rather general and occurred with many genes, but below the initial cutoff of occurrence with 100 genes.

LSA in MeSH Term Space for 3 Gene Clusters

The SVD/LSA was performed on the resulting IGF weighted gene-MeSH term frequency matrices for the three co-expression clusters. Figure 5.10 shows the variance captured by the first 50 SVD modes of the gene-MeSH term matrices for the three clusters. Cluster 1 corresponds to the red cluster in Figure 5.6, cluster 2 to the green cluster and cluster 3 to the blue cluster in Fig. 5.6. Also shown, by red lines, are the variances found by SVD in 5 random groups of genes of the same size

[11] The Euclidean length of the SVD singular vectors is normalized to one, and a MeSH term with a coefficient of 0.7 or more therefore determines half $(0.7^2 = 0.49)$ or more of the length of that singular vector.

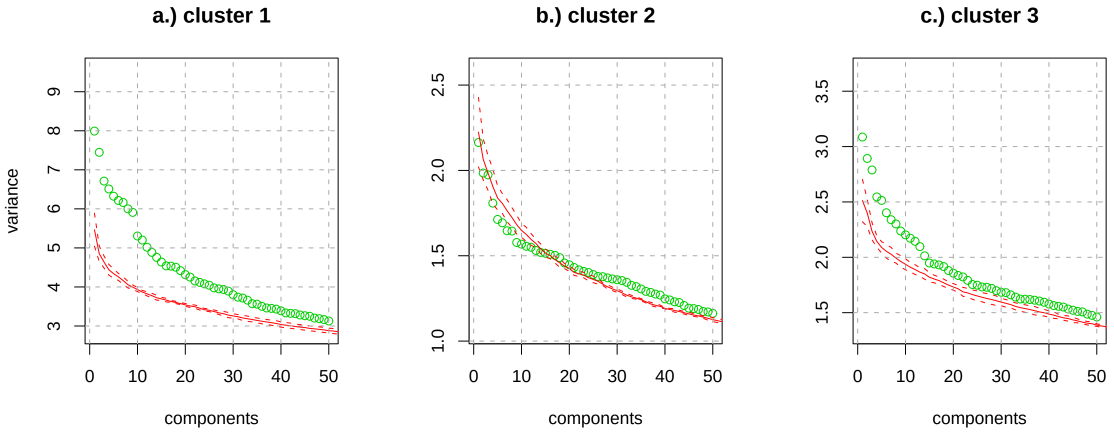


Figure 5.10: The variance for the first 50 components of the SVD of the gene-MeSH association matrices for the three co-expression clusters and for random groups of genes. The green point plots show the variances for the respective gene cluster, the continuous red lines show the average variance for 5 random groups of genes (with the same number of genes as the respective cluster). The standard deviation from the mean for the 5 random groups is shown by the dashed lines. Cluster 1 corresponds to the red cluster in Fig. 5.6. Cluster 2 corresponds to the green cluster and cluster 3 to the blue cluster in Fig. 5.6.

as the clusters (note, these 15 random sets of genes were independent of the sets used to remove outliers). The variances for the co-expression clusters 1 and 3 show increased variances over the random groups of genes. For cluster 2, however, the variances of the first few modes are not higher than for same sized random groups of genes.

Figure 5.11 shows the projection of the genes (black circles) and MeSH terms (red triangles) onto the respective singular vectors of modes 1 and 2 (left figure) and modes 3 and 4 (right figure). Table 5.2 lists the 20 genes most correlated with mode 1 and anti-correlated with mode 2. The functional theme of this group of genes could be called “small nuclear ribonucleoproteins and splicing factors”. The MeSH terms most associated with these genes are “Spliceososmes”, “RNA, Small Nuclear”, “ Ribonucleoprotein, U1 Small Nuclear” and “Ribonucleoprotein, U2 Small Nuclear”. As small nuclear ribonucleoproteins are frequently part of spliceosomes and splicing factors, the MeSH terms and genes are clearly related. The genes most correlated with mode 1 and positively correlated with mode 2 split into three distinct groups in the subspace of modes 3 and 4: genes positively correlated with mode 4 and negatively with mode 3 (Table 5.3), genes correlated negatively with both modes 3 and 4 (Table 5.4), and genes positively correlated with mode 3 (and mostly uncorrelated with mode 4) (Table 5.5). Genes of Table 5.3 are mostly transcrpition factors, some associated with regulating the cell-cycle. Genes of Table 5.4 are overwhelmingly associated with Thyroid receptor proteins or proteins binding to the receptor. Table 5.5 contains a significant

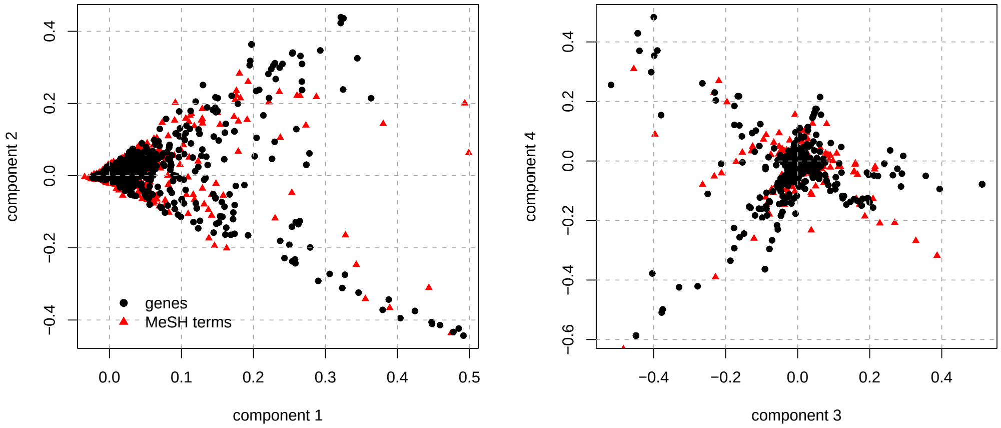


Figure 5.11: Projection of genes (black circles) and MeSH terms (red triangles) onto the first two SVD components (left) and components 3 and 4 (right) for cluster 1.

number of transcription factors (different from Table 5.3) and Retinoblastoma binding proteins.

For cluster 2 genes, the same analysis was performed. Figure 5.12 shows fewer genes were found to be strongly correlated with the first 2 modes than for cluster 1 (as might be expected from the singular value spectrum). Genes (anti-) correlated with mode 1 were grouped into genes correlated with mode 2 and anti-correlated with mode 2. Both groups contain mainly immune system regulating proteins: interferon response genes, signal transduction and apoptosis related proteins in Table 5.6 and inflammatory cytokines, as well as cell adhesion and apoptosis related proteins in Table 5.7.

Three groups of genes for cluster 3 were found correlated with the first two modes (Figure 5.12). Table 5.8 lists proteins correlated with mode 1. Many extracellular and cell adhesion related proteins are found in this group of genes. Genes negatively correlated with mode 2 (Table 5.9) list extracellular matrix proteins, many of them related to Collagen. Several proteins in Table 5.10 are related to membrane channel proteins and dehydrogensases.

5.4 Discussion

We compare the functional groups of genes we identified for the 3 co-expression clusters to human annotations in [Challacombe et al., 2004] (also see chapter 3) and [Browne et al., 2001]. The functional annotations of genes in clusters 1 and 2 were manually inspected and results reported in chapter 3 and [Challacombe et al., 2004]. [Challacombe et al., 2004] reports a “noticeably greater percentage of genes in cluster 1 in the categories of transcription and oncogenesis/cell cycle regula-

O75533	Splicing factor 3B subunit 1; Spliceosome associated protein 155
Q15459	Splicing factor 3 subunit 1; Spliceosome associated protein 114
Q15427	Pre-mRNA splicing factor SF3b 49 kDa subunit; Spliceosome associated protein 49
Q13435	Pre-mRNA splicing factor SF3b 145 kDa subunit; Spliceosome associated protein 145
P43331	Sm-D3; Small nuclear ribonucleoprotein Sm D3; snRNP core protein D3
P08579	U2 small nuclear ribonucleoprotein B
Q15356	Sm-F; Small nuclear ribonucleoprotein F
Q15357	Sm-G; Small nuclear ribonucleoprotein G
P14678	Sm-B/Sm-B”; Small nuclear ribonucleoprotein associated proteins B and B”
P09661	U2 small nuclear ribonucleoprotein A”; U2 snRNP-A”
Q14562	ATP-dependent helicase DDX8; RNA helicase HRH1; DEAH-box protein 8
Q07955	pre-mRNA splicing factor SF2, P33 subunit; Alternative splicing factor ASF-1
P08621	U1 small nuclear ribonucleoprotein 70 kDa; U1 snRNP 70 kDa
Q9Y4Y8	U6 snRNA-associated Sm-like protein LSm6
O43143	ATP-dependent RNA helicase #46; Putative pre-mRNA splicing factor RNA helicase; DEAH box protein 15
Q14498	RNA-binding region containing protein 2; Splicing factor HCC1
O14893	Survival of motor neuron protein-interacting protein 1; Gemin2
O00566	U3 small nucleolar ribonucleoprotein protein MPP10; M phase phosphoprotein 10
Q13487	snRNA activating protein complex 45 kDa subunit; Proximal sequence element-binding transcription factor delta
Q16533	snRNA activating protein complex 43 kDa subunit; Proximal sequence element-binding transcription factor gamma

Table 5.2: The SwissProt accessions and names for the genes in cluster 1 correlated positively with mode 1 and negatively with mode 2 (see Figure 5.11; the genes are ordered by decreasing correlation with mode 1; synonym names are separarted by a semicolon). Most genes are mRNA splicing factors and ribonucleoproteins.

Q16514	TAFII-20/TAFII-15; Transcription initiation factor TFIID 20/15 kDa subunits
O14981	TAF(II)170; TBP-associated factor 172
Q15544	TAFII-28; Transcription initiation factor TFIID 28 kDa subunit
Q15543	TAFII-18; Transcription initiation factor TFIID 18 kDa subunit
O43513	Cofactor required for Sp1 transcriptional activation subunit 9; Transcriptional co-activator CRSP33
O00268	TAFII-130
Q15545	TAFII-55
Q15542	TAFII-100; Transcription initiation factor TFIID 100 kDa subunit
P52657	TFIIA-12; Transcription initiation factor IIA gamma chain
P13984	TFIIF-beta; Transcription initiation factor RAP30
P20226	TATA box binding protein; Transcription initiation factor TFIID
Q00403	Transcription initiation factor IIB
P51948	CDK-activating kinase assembly factor MAT1; Cyclin G1 interacting protein
P52655	TFIIA-42; Transcription initiation factor IIA alpha and beta chains
P51946	Cyclin H; MO15-associated protein
P32780	TFIIH basal transcription factor complex p62 subunit

Table 5.3: Cluster 1: genes positively associated with mode 2, negatively with mode 3 and positively with mode 4, ordered by decreasing correlation with mode 2. Most of the proteins belong to the group of the so called “general transcription factors” that bind to RNA Polymerase II and that are required to initiate transcription.

Q15648 Peroxisome proliferator-activated receptor binding protein; Thyroid receptor interacting protein 2; p53 regulatory protein RB18A

Q9UHV7 Thyroid hormone receptor-associated protein complex 240 kDa component; Trap240

Q09472 E1A-associated protein p300

Q15649 TRIP-3; Thyroid receptor interacting protein 3

Q15643 TRIP-11; Thyroid receptor interacting protein 11

Q14669 TRIP-12; Thyroid receptor interacting protein 12

Q15650 TRIP-4; Activating signal cointegrator 1; Thyroid receptor interacting protein 4

P47210 26S protease regulatory subunit 8; Proteasome subunit p45; TRIP-1; Thyroid hormone receptor interacting protein 1

Q15642 Cdc42-interacting protein 4; TRIP-10; Thyroid receptor interacting protein 10

P35790 CHETK-alpha; Choline kinase

Q14686 Nuclear receptor coactivator 6; Peroxisome proliferator-activated receptor-interacting protein; Cancer-amplified transcriptional coactivator ASC-2; Thyroid hormone receptor-binding protein

P18583 Protein C21orf50; Negative regulatory element-binding protein

Q99963 EEN-B2; SH3-containing GRB2-like protein 3

Q9Y3I1 F-box only protein 7

O43504 HBV X interacting protein; Hepatitis B virus X interacting protein

P41002 G2/mitotic-specific cyclin F

Table 5.4: Cluster 1: genes positively associated with mode 2, negatively with mode 3 and negatively with mode 4, ordered by decreasing correlation with mode 2. The thyroid-hormone receptors are hormone-dependent transcription factors that control expression of many target genes [Park et al., 1993].

O75461	Transcription factor E2F6
Q14186	Transcription factor DP-1; E2F dimerization partner 1
O75367	Core histone macro-H2A.1
Q13185	HECH; Chromobox protein homolog 3
Q14493	Histone RNA hairpin-binding protein
P17317	H2A/z; Histone H2A.z
Q09028	Chromatin assembly factor 1 subunit C; CAF-1 subunit C; Retinoblastoma binding protein 4
Q06587	Polycomb complex protein RING1; RNF1
O00716	Transcription factor E2F3
Q15291	Retinoblastoma-binding protein 5; RBBP-5
P29374	Retinoblastoma-binding protein 1; RBBP-1
P29375	Retinoblastoma-binding protein 2; RBBP-2
Q01094	Retinoblastoma binding protein 3; RBAP-1
O96020	G1/S-Specific cyclin E2
P24864	G1/S-specific cyclin E1
P06400	Retinoblastoma-associated protein; RB
Q15329	Transcription factor E2F5; E2F-5
Q08999	Retinoblastoma-like protein 2; RBR-2
P32519	ETS-related transcription factor Elf-1

Table 5.5: Cluster 1: genes positively associated with mode 2 and mode 3, ordered by decreasing correlation with mode 2. Many genes in this cluster are transcription factors and transcriptional regulators involved in oncogenesis and cell cycle regulation some are involved in apoptosis [Challacombe et al., 2004].

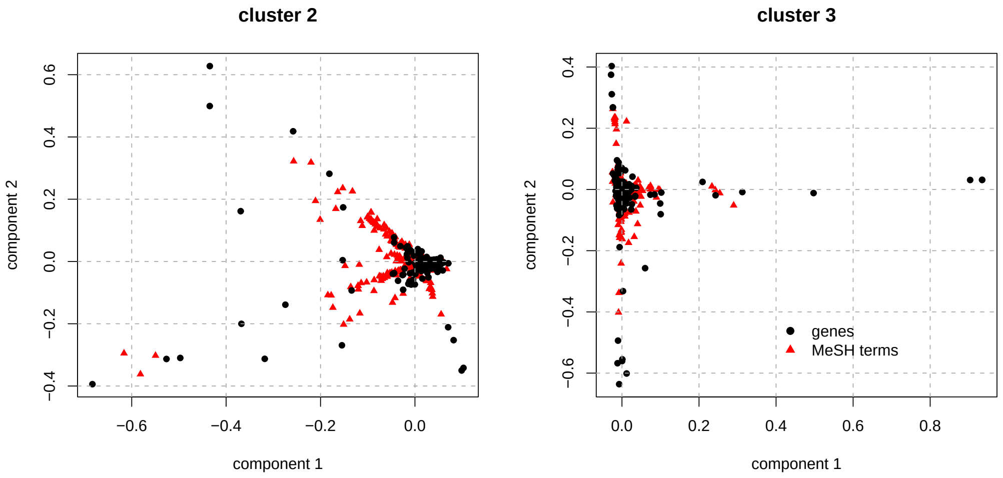


Figure 5.12: Projection of genes (black circles) and MeSH terms (red triangles) onto the first two SVD components for cluster 2 cluster 3. Fewer genes than for cluster 1 are found associated with the respective LSA modes.

Q13651 IL-10R1; Interleukin-10 receptor alpha chain precursor

Q08334 IL-10R2; Interleukin-10 receptor beta chain precursor

P01579 Interferon gamma precursor; IFN-gamma; Immune interferon

P42701 IL-12RB1; Interleukin-12 receptor beta

P80217 Interferon-induced 35 kDa protein; IFP 35

P51692 Signal transducer and activator of transcription 5B

P52198 Rnd2; Rho-related GTP-binding protein RhoN

P25446 Tumor necrosis factor receptor superfamily member 6 precursor; FASL receptor; Apoptosis-mediating surface antigen FAS; CD95

P20290 RNA polymerase B transcription factor 3; Transcription factor BTF3

P19075 Tumor-associated antigen CO-029

Table 5.6: Cluster 2: genes anti-correlated with mode 1 and mode 2. Interferon response genes (immune system regulation), signal transduction, apoptosis related proteins.

P10147	Small inducible cytokine A3 precursor; Macrophage inflammatory protein 1-alpha; G0/G1 switch regulatory protein 19-1; PAT 464.1
P13236	Small inducible cytokine A4 precursor; Macrophage inflammatory protein 1-beta; T-cell activation protein 2
P25024	IL-8R A; High affinity interleukin-8 receptor A; IL-8 receptor type 1
P51685	CC-chemokine receptor CHEMR1
P80098	Small inducible cytokine A7 precursor; Monocyte chemotactic protein 3
P32302	C-X-C chemokine receptor type 5; MDR15; Monocyte-derived receptor 15
P30740	EI; Monocyte/neutrophil elastase inhibitor
Q9NRI5	Disrupted in schizophrenia 1 protein
Q14289	Related adhesion focal tyrosine kinase; Cell adhesion kinase beta
Q14790	Caspase-8 precursor; Apoptotic protease Mch-5

Table 5.7: Cluster 2 : genes anti-correlated with mode 1 and positively correlated with mode 2. Mostly inflammatory cytokines, also cell adhesion and apoptosis related proteins.

tion than in cluster 2". Our analysis revealed three groups of transcription factors that are strongly associated with the first few LSA modes in cluster 1, including cell-cycle and oncogenesis regulators in Table 5.5. Analysis in chapter 3 also lead to the conclusion that "cluster 2 contained a higher percentage of genes involved in signal transduction, immune system regulation, and cell adhesion compared to cluster 1". The two first LSA modes of cluster 2 genes mainly contained immune system regulating and signal transduction proteins (many interferon response genes and inflammatory cytokines), as well as some apoptosis and cell adhesion related proteins. This corresponds also very well to the finding of [Browne et al., 2001], who report a decrease in expression in the first 8 hrs post infection for interferon response genes and inflammatory cytokines (genes in our cluster 2 show a decrease in expression between 4 and 20 hours post infection). All functional groups of genes that are reported for cluster 1 and 2 in [Challacombe et al., 2004], as well as in [Browne et al., 2001], are found among the first few LSA modes in our analysis for the respective clusters (e.g. cell-cycle and oncogenesis transcription factors, immune system regulators, apoptosis and cell adhesion related proteins). Interestingly, we also found some functional groups of genes in clusters 1 that were not reported in [Challacombe et al., 2004] nor in [Browne et al., 2001]. The largest not reported group of genes are small ribonucleoproteins and splicing factors in cluster 1 (Table 5.2). This group of genes is important for the processing of the host messenger RNA before transport to the cytoplasm and translation to proteins. A survey of the literature revealed that the herpes virus can severely impact the host cell's protein production by interfering with the splicing of the host mRNA. The literature typically reported a direct interaction of the virus proteins with

P07942 Laminin beta-1 chain precursor

P11047 Laminin gamma-1 chain precursor

Q16363 Laminin alpha-4 chain precursor

P98160 PLC; Basement membrane-specific heparan sulfate proteoglycan core protein precursor

P02545 70 kDa lamin; Lamin A/C

P31431 Ryudocan core protein; Amphiglycan; Syndecan-4 precursor

P18827 CD138 antigen; Syndecan-1 precursor

P47914 60S ribosomal protein L29; Cell surface heparin binding protein HIP

P13611 Versican core protein precursor; Large fibroblast proteoglycan; Chondroitin sulfate proteoglycan core protein 2

O94766 GlcUAT-I; Glucuronosyltransferase-I

P16070 CDw44; Heparan sulfate proteoglycan; GP90 lymphocyte homing/adhesion receptor; Extracellular matrix receptor-III; CD44 antigen precursor

P27544 LAG1 protein; Embryonic growth/differentiation factor 1 precursor; Longevity assurance homolog 1

P29279 Hypertrophic chondrocyte-specific protein 24; Connective tissue growth factor precursor

P36956 Sterol regulatory element binding protein-1

Q12772 Sterol regulatory element binding protein-2

Q16394 Putative tumor suppressor protein EXT1

Table 5.8: Cluster 3: genes positively correlated with mode 1. Many extracellular and cell adhesion related proteins. Laminin is a large, noncollagenous glycoprotein with antigenic properties. It functions to bind epithelial cells to the basement membrane (MeSH annotation [National Library of Medicine, 2004]).

P05997	Collagen alpha 2(V) chain precursor
P20849	Collagen alpha 1(IX) chain precursor
P02461	Collagen alpha 1(III) chain precursor
Q07092	Collagen alpha 1(XVI) chain precursor
P12109	Collagen alpha 1(VI) chain precursor
P08123	Collagen alpha 2(I) chain precursor
O94833	Dystonia musculorum protein
P29279	Connective tissue growth factor precursor; Hypertrophic chondrocyte-specific protein 24
P07996	Thrombospondin 1 precursor
P35555	Fibrillin 1 precursor
P13611	Versican core protein precursor; Large fibroblast proteoglycan
Q14192	Skeletal muscle LIM-protein 3
P22003	Bone morphogenetic protein 5 precursor
P35442	Thrombospondin 2 precursor
P35556	Fibrillin 2 precursor

Table 5.9: Cluster 3: genes negatively correlated with mode 2. Extracellular matrix proteins. Collagen is the main constituent of skin, connective tissue and the organic substance of bones and teeth.

O75783	Rhomboid-like protein 1
O75154	Eferin
P29372	N-methylpurine-DNA glycosirase
O95180	Voltage-dependent T-type calcium channel alpha-1H subunit
P50550	P18; Ubiquitin-conjugating enzyme UbcE2A; SUMO-1-protein ligase
P98161	Polycystin 1 precursor
P22674	Uracil-DNA glycosylase 2
P34969	5-hydroxytryptamine 7 receptor; Serotonin receptor
P15382	Potassium voltage-gated channel subfamily E member 1
Q9BYH1	Seizure 6-like protein precursor
Q03135	Caveolin-1
Q92952	SK1; Small conductance calcium-activated potassium channel protein 1
Q12809	eag homolog; Potassium voltage-gated channel subfamily H member 2
Q8TDN2	Potassium voltage-gated channel subfamily V member 2
P00325	Alcohol dehydrogenase beta chain
P00326	Alcohol dehydrogenase gamma chain
P11766	FDH; Alcohol dehydrogenase class III chi chain (EC 1.1.1.1)
Q01959	Sodium-dependent dopamine transporter
O75828	Carbonyl reductase [NADPH] 3

Table 5.10: Cluster 3: genes positively correlated with mode 2.

splicing factors [Hardy and RM., 1994], but our analysis suggests that the transcription of many splicing factors of the host cell is affected as well. In fact, cluster 1 genes are up-regulated genes. The host cell might respond to the interference of the virus proteins with host mRNA processing by increasing the production of ribonucleoproteins and splicing factors, to improve host mRNA processing. This reasoning might also explain the finding that the "general transcription factors" in Table 5.3 are upregulated. One reason why this group of genes was not identified in cluster 1 by the human expert (chapter 3 and [Challacombe et al., 2004]) might be because the expression of this functional group of genes was not expected to be affected. The human annotator states that she focused on functional classes of genes whose transcription is known to be influenced by the virus: signal transduction, immune system regulation, apoptosis, cell cycle regulation, oncogenesis, cell adhesion and transcription. Ribonucleoproteins and splicing factors do not fall within these functional classes, therefore they were missed in the analysis. This illustrates the potential value of the exploratory, inference driven functional data mining approach applied here. Another functional group of genes not explicitly mentioned in [Challacombe et al., 2004] or [Browne et al., 2001] are the Thyroid hormone receptor transcription factors in Table 5.4. Research of [Park et al., 1993] suggests that regulating thyroid hormone receptor expression may play an important role in regulating the life cycle of the herpes simplex virus in the host cell. Cluster 3 genes were not annotated by the human expert in [Challacombe et al., 2004] and no comparison to our findings could be made. The singular value spectrum for this cluster in MeSH term space was above that of random groups of genes for the first few modes. A significant number of genes in cluster 3 are cell adhesion molecules and extracellular proteins, e.g. laminin, cell surface glycoproteins and collagen. [Challacombe et al., 2004] reports that cell adhesion molecules are key to several functions of the immune response, including T cell-antigen-presenting cell interactions, T cell-B cell interactions, and cytotoxic T cell/NK cell interactions with the infected target cells. All of these are essential components for the generation of effective inflammatory responses and the development of rapid immune responses. Genes in cluster 3 are repressed in their expression in the later time points. It is therefore likely that the virus inhibits the expression of these cell adhesion molecules to inhibit the host's immune response. Potassium, Sodium and Calcium levels have been reported to be affected by HCMV and other herpes virus infections [Hackstadt and Mallavia, 1982,Browne et al., 2001], which could explain the group of channel proteins we identified in cluster 3 and listed in Table 5.10.

In conclusion, we demonstrated the potential value of literature mining, here specifically the mining with MeSH terms, for functional information. We were able to validate our findings with what had been found previously by experts and their manual evaluation of annotation data. In addition, we identified new functional groups of genes in the co-expression clusters that had not been reported in these expert studies, probably because the focus of the manual evaluation of annotation data was on different functional groups. We do not claim that our approach can replace expert inspection of the data. The functional information reported in [Browne et al., 2001, Challacombe et al., 2004] was more detailed than our automated analysis could provide. What our analysis can provide, though, are "functional themes" for groups of genes and proteins that can guide the expert annotator and focus his or her work. In addition, our methodology might point to functional themes and groups of genes that are not expected and might be missed in the large amounts of data when dealing with hundreds or even thousands of genes.

Chapter 6

Pfam Protein Family Prediction in MeSH Space[1]

6.1 Introduction

Mining of biological information from databases and literature gains increasing importance as both the amount of data from high-throughput experiments and the amount of biological knowledge stored in databases and literature increases. Different techniques for information mining in Bioinformatics have been presented but usually in very specific and different contexts, gene network inference from literature data, functional annotation of proteins, and improvement of remote homolog detection for proteins [Masys et al., 2001, Jenssen et al., 2001, Andrade and Valencia, 1998, MacCallum et al., 2000]. What has been missing in the field are large-scale studies that allow for quantitative validation and a gold standards defining an effective basis for method comparison. Here we propose such a large-scale, quantitative approach for evaluation and comparison of methods for information retrieval for Bioinformatics from literature. The large scale test set against which we test our literature mining approach is the Pfam protein sequence classification [Sonnhammer et al., 1997, Bateman et al., 2004]. Pfam is a manually curated collection of protein families, currently encompassing several thousands of families. Genome projects, including both the human and fly, have used Pfam for large scale functional annotation of genomic data. The proteins of a Pfam family are functionally very similar due to their similarity in sequence. It is this congruence of Pfam with protein functional classes, as well as its classification based on a physical property of proteins, their sequence, that makes it an ideal test set for objective evaluation and comparison of information retrieval and knowledge discovery mining algorithms in Bioinformatics.

The specific knowledge discovery approach we test here is the vector space model [Manning

[1] Submitted to ISMB/BioLink 2005 *[Rechtsteiner et al., 2005]*.

and Schütze, 1999] of Information Retrieval in combination with the biomedical indexing vocabulary MeSH (Medical Subject Heading Vocabulary). The National Library of Medicine (NLM) uses MeSH to index all the biomedical publications in its literature database MEDLINE[2]. MeSH is a controlled, hierarchically organized vocabulary that has been developed and adapted to new knowledge domains by NLM for decades. MeSH contains over 22,000 terms and 100,000 synonyms (so-called entry terms). The algorithm we use here to represent and discover knowledge in the MeSH vocabulary is the vector space model. Each protein will be represented by a MeSH term vector, obtained from the literature about that protein. A similarity measure can then be defined for proteins in that MeSH term space. If two proteins are functionally related, and the literature and MeSH indexing terms capture the functional information about the proteins, we expect the MeSH term vectors for the proteins to be similar. For functionally very different proteins, we expect the MeSH term vectors to be different. As Pfam families are functionally congruent, i.e. proteins in a family are functionally closely related, we expect Pfam families to cluster in MeSH term space. To test this hypothesis, we take a protein's Pfam family to be unknown and classify it into a Pfam family based on its neighbor proteins in MeSH term space and their Pfam families. If our hypothesis is correct, and publications about proteins and the corresponding MeSH indexing terms capture functional information about proteins, this classification should be successful in most cases. Further, we can assess how well the corresponding MeSH vectors describe proteins and their functions.

Our study contains 15,217 proteins from 1611 Pfam families. If knowledge discovery techniques are supposed to be useful for the increasingly large-scale studies and data sets in Bioinformatics, they need to be able to perform well on such large data sets and need to be tested and compared on such. A technique that works for few functional classes (e.g. see the studies by [Masys et al., 2001, Andrade and Valencia, 1998]), for example for the separation of two groups of proteins with very different functions, might not work for the separation of many functional groups when the resolution of functional differences needs to be at a more detailed level. But exactly this is the challenge for the future of information mining and knowledge discovery in Bioinformatics. We also needed for our study a body of publications that are associated with and are about the proteins from the 1611 Pfam families. We obtained these publications from the SwissProt/UniProt [SIB/EBI, 2004] protein sequence database. SwissProt is a manually curated database and the information it contains, e.g. literature references for the respective protein sequences, is therefore very reliable. For the 15,217 proteins we obtained 26,411 publications from SwissProt. From the literature database MEDLINE the MeSH indexing terms for these publications were obtained.

[2] In fact, unless specified otherwise, any query text string that is entered in PubMed (the WWW gateway of MEDLINE [National Library of Medicine, 2005]) is first mapped to MeSH terms and the respective documents indexed by these terms are then retrieved, ordered by some significance score.

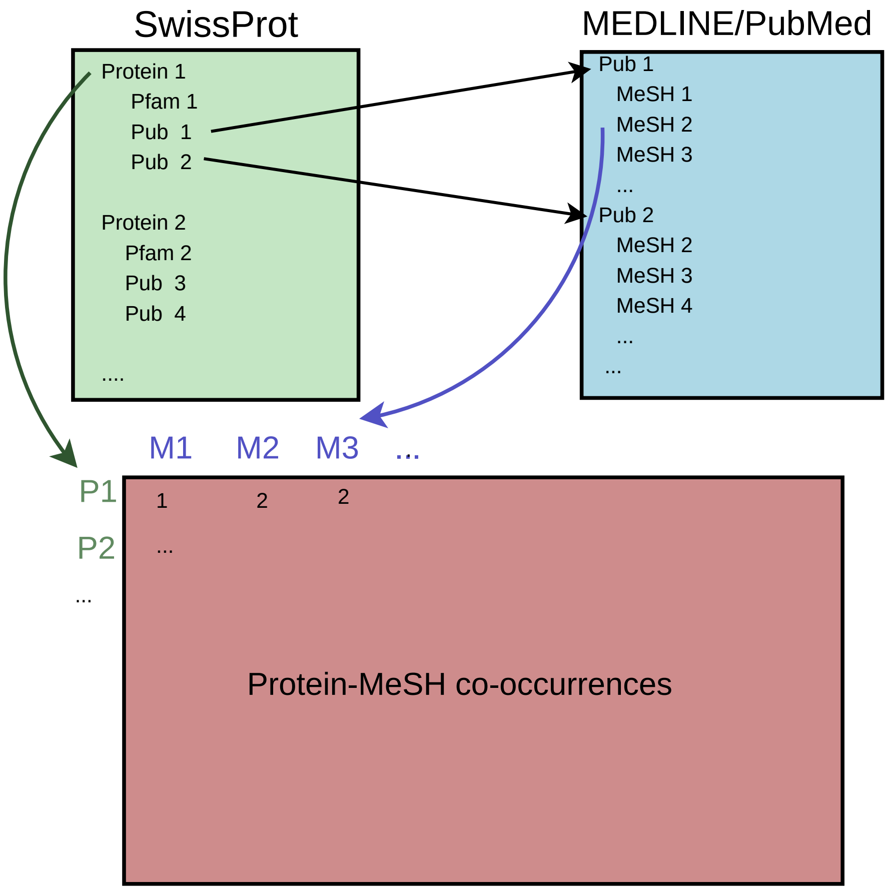


Figure 6.1: Our data was obtained from the SwissProt protein sequence database and the MEDLINE/PubMed literature database. SwissProt is a protein sequence database curated by experts. Besides the amino acid sequence of a protein it also lists different types of annotations, cross-references to other databases, e.g. the Pfam family of a protein, and references to relevant publications for the protein. The publication references were mapped to the respective publications in the biomedical literature database MEDLINE. From there we obtained the MeSH indexing terms for the publications of each protein. This information can then be represented in a protein-MeSH co-occurrence table, where the entry for a given protein-MeSH term pair indicates the number of publications referenced by the protein and indexed by the MeSH term. The proteins, represented by the rows of this co-occurrence table, can be interpreted as vectors in MeSH term space (some weighting factor is typically applied to the term dimensions, as discussed in section 6.2).

For successful separation and prediction of Pfam families, NLM's manual indexing of publications with MeSH needs to be performed well and consistently. If different indexers chose vastly different indexing terms for proteins from the same Pfam family, prediction of the correct Pfam family for a protein will be difficult. Our study indirectly sheds some light on this question of MeSH indexing consistency. The study of Funk et al. [Funk and Reid, 1983] reported a 40-60% overlap of MeSH terms assigned by different indexers to the same publication. Our study will illuminate if such overlap is sufficient to separate and predict Pfam families.

Another question we explored is the one of synonymy and polysemy in the MeSH term vocabulary. It has been shown that the performance of the vector space model in information retrieval can be improved significantly by identifying with Singular Value Decomposition (SVD) the subspaces in the term-document space with highest variances. This technique is referred to as Latent Semantic Indexing (LSI) or Latent Semantic Analysis (LSA) [Deerwester et al., 1990, Berry et al., 1995]. LSI detects correlations among terms in the set of documents and therefore can weaken negative effects of term synonymy (multiple terms have the same meaning) and term polysemy (terms that have multiple meanings). We applied LSA to our protein-MeSH co-occurrence matrix and predicted Pfam families in the resulting reduced SVD spaces. This technique provided very little improvement over predicting in the original protein-MeSH term space, and only with a thousand or more dimensions (singular vectors). This indicates that there is little synonym or polysemy in the MeSH vocabulary, which can be expected from a well-designed, controlled vocabulary.

Related work to ours was presented by Andrade et al. [Andrade and Valencia, 1998]. They took 71 functional groups of proteins and extracted keywords from abstracts (versus using MeSH terms) from publications referenced in SwissProt. The distribution of this “bag of keywords” over the families served as a background distribution against which they compared keywords from a new protein or protein family. Given a new protein or family and its literature, keywords are identified that occur significantly more often in the literature of this new protein or family than in the background distribution. The presented technique was validated with anecdotal evidence and only with a few example proteins. The number of protein families and body of literature was small, our data set is significantly larger (1611 families). As mentioned before, an approach that works for few, very different functional groups, might not work for many families where the “function space” is more “crowded”, i.e. there is overlap in the functions of the families. But if a literature mining approach is supposed to be of value for the increasing large scale tasks in Bioinformatics, it needs to scale well to such larger scopes.

An application of information mining for gene co-expression clusters using MeSH terms has been presented by Masys et al. [Masys et al., 2001]. Their study has taken a small set of publications associated with two groups of differently expressed genes in two different medical conditions (two different blood leukemia). They then identified the MeSH terms that occurred significantly more often with the respective groups of genes. They showed that the identified MeSH terms were informative about the gene groups and the two medical conditions with which their increased expression was associated. The study again focused on only few (two) groups of functionally different entities and the validation was again heuristic for these two groups.

Much of the current information mining work in Bioinformatics is still performed for very specific tasks with often rather small scopes. The results are often validated with anecdotal evidence relating to the task at hand, e.g. the specific medical conditions a gene expression data set was obtained for. We test our method, literature set and MeSH against a large test data set, the Pfam family classification, which is based on an objective, physical property of proteins, their sequence similarity. We are testing if the large set of Pfam families and the literature MeSH term space are mutually coherent. As Pfam is often congruent with functional classes, our study and its results suggest how well our method should perform in tasks other than Pfam classification, for example the prediction of function for groups of proteins or genes.

6.2 Methods and Data

6.2.1 Data

The literature data set for this study were the publications referenced by proteins in SwissProt and the MeSH terms for the publications were obtained from MEDLINE (see Figure 6.1). SwissProt is manually curated by experts and this set of publications can therefore be considered an *expert literature* set (see also [Shatkay et al., 2000]). In the work here SwissProt version 41.0 was used, which contains 122,564 protein sequence entries. 89,143 of these had Pfam family references (3938 different Pfam families) as well as 75,649 distinct publication references. As we wanted to establish a baseline in this study, we performed several filtering steps to eliminate any “artificial links” between proteins and Pfam families as well as to remove “noise” from the data set.

First, the 15% of proteins were removed that had more than one Pfam family referenced. These proteins would have linked the respective Pfam families[3]. Next we filtered out all publications that were referenced by multiple proteins from more than one Pfam family. Many of these publications are about sequencing, e.g. of chromosomes or whole genomes and therefore can be referenced by many proteins. Such publications will not contain specific information about proteins and their families and therefore will add noise to the literature data. 47,368 MEDLINE publications, or 63%, were referenced by proteins from only one Pfam family and they were retained in the literature data set. The number of proteins for which we still had literature references after this filtering

[3] Such links might be desirable, as Pfam families occurring together in proteins might often be functionally related. But in this baseline study we wanted to detect any functional relationships of proteins and Pfam families purely from independent literature.

step decreased however to 32,922 (43% of the proteins that had one Pfam family reference). This suggests that many of the protein sequences in SwissProt have, to this date, literature references that are not specific to the protein or its family and their functions but are of a more general type. Two more filtering steps were applied to the data. First, all proteins that had exactly identical literature references were filtered out except for one representative (selected at random). This step removed mainly proteins that were identical or closely related but, for example, from different organisms. The next filtering step removed literature references to the same publication by multiple proteins (of the same Pfam family) except for the reference by one of the proteins (we chose the protein that had the fewest literature references). Both of the last two filtering steps insure that any similarity in MeSH term space (e.g. similar MeSH term vectors) of proteins from the same Pfam family are due to independent and not shared publications.

The above filtering steps are conservative and control for any possible artificial link of proteins in MeSH space. The publications for each protein (and therefore the respective MeSH terms) are independent from each other. Any relationships detected in our study among proteins and Pfam families are due to independent literature and therefore are due to the related information contained in the publications of the respective proteins and Pfam families.

After filtering, the data set contained 27,682 proteins, referencing 47,368 MEDLINE publications (each publication only referenced once, by a single protein) and 2503 Pfam families. On average, each protein references 1.7 publications and each Pfam family has 13 protein members. 892 or 36% of all Pfam families have only 1 or 2 proteins, 1611 or 64% have 3 or more proteins. 296 Pfam families, or 12%, have 20 or more proteins[4]. Due to the nature of our classification algorithm (discussed in detail later) we predicted Pfam families only for the proteins of the 1611 Pfam families with 3 or more protein members[5]. To limit the bias of our classification algorithm towards larger families (see also discussion later), we limited the size of Pfam families to 20 protein members. For Pfam families with more than 20 protein members, 20 were selected at random. This lead to a data set with 15,217 proteins, from 1611 Pfam families with 26,411 publications and 5,639 different MeSH terms[6].

[4] The 5 largest families are the 7 transmembrane receptor rhodopsin family PF00001 (Pfam id) with 810 proteins, the Immunoglobulin domain PF00047 with 525 proteins, the Globin family PF00042 with 499 proteins, the Protein kinase domain PF00069 with 494 proteins and the Homeobox domain PF00046 with 372 proteins.

[5] In principle our nearest neighbor classification algorithm could also have predicted families with 2 protein members. However, the performance for small families decreases fast and we selected a cutoff of 3.

[6] ~2000 MeSH terms that occurred with only one protein were removed. These MeSH terms do not link any proteins.

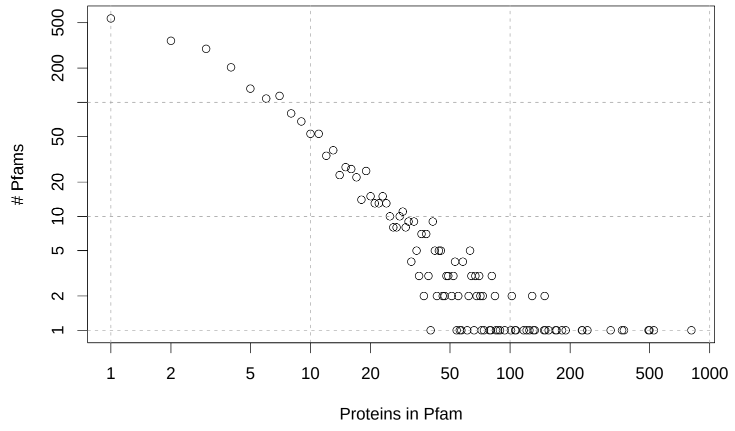


Figure 6.2: Distribution of the 27,682 proteins over the 2503 Pfam families. 892 Pfam families (36%) have only 1 or 2 proteins, 1611 have 3 or more proteins. 296 Pfam families, or 12%, have 20 or more proteins.

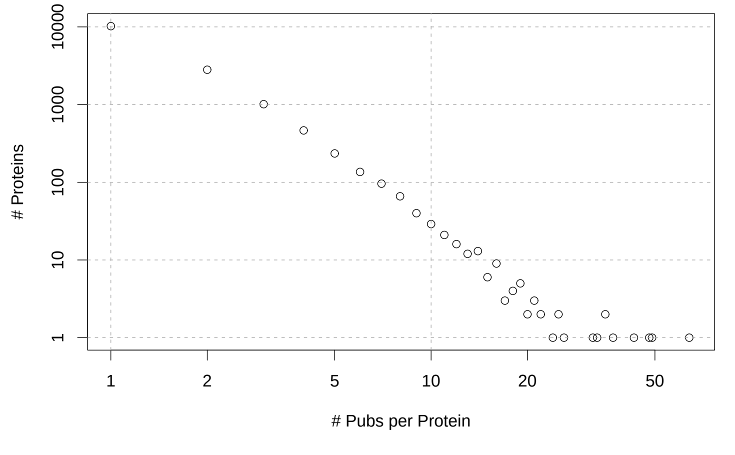


Figure 6.3: Distribution of 26,411 publications over the 15,217 proteins from 1611 Pfam families (with size 3 or more protein members). The average number of different publications referenced per protein is 1.7. But 67% of the proteins (10,220) have only 1 document. An additional 18% have 2 documents. 97% (14,741) have 5 or less documents. One protein has 64 documents[8].

6.2.2 The Vector Space Model in Information Retrieval

The vector space model in Information Retrieval (IR) represents documents in (typically high-dimensional) keyword space [Manning and Schütze, 1999, Baeza-Yates et al., 1999]. Here we have adapted that model to represent proteins in MeSH keyword space (see also Fig. 6.4 for an illustration). Each coefficient of the protein vector in MeSH space is made up of what is called a *local weight* which is then multiplied by a *global weight*. The local weight is typically referred to as the term frequency tf_{ik}. Here tf_{ik} is the number of publications cited for protein i in SwissProt that are also indexed by MeSH term k in MEDLINE. The global weight, here denoted idf_k represents a weighting of the MeSH term dimension k, which is supposed to reflect the information content of MeSH term k. The global weighting applied in this study is discussed in the next subsection.

The coefficient of protein vector i in MeSH term dimension k is then given by $w_{ik} = tf_{ik} * idf_k$. w_{ik} is the k^{th} coefficient of the protein vector $\mathbf{p}_i$. We used as the similarity measure between protein vectors in MeSH space the cosine measure (a common distance measure for the vector space model in IR [Baeza-Yates et al., 1999, Manning and Schütze, 1999, Deerwester et al., 1990, Berry et al., 1995]): given protein vectors $\mathbf{p}_i$ and $\mathbf{p}_j$ in n-dimensional term space, the cosine between these protein vectors is given by the normalized dot product:

$$cos(\mathbf{p_i}, \mathbf{p_j}) = \frac{\mathbf{p_i p_j}}{|\mathbf{p_i}||\mathbf{p_j}|} \tag{6.1}$$

6.2.2.1 Global MeSH term weighting

The most popular weight in IR is the Inverse Document Frequency (IDF) [Dumais, 1990, Manning and Schütze, 1999]. The weighting factor idf_k for a term k is defined as $idf_k = log(\frac{N}{n_k})$ where N is the total number of documents in the collection and n_k is the number of documents indexed by term k. Note that if term k indexes every document it will have no information to discriminate among the documents. This is reflected in the IDF weighting: $n_k = N$ and therefore $idf_k = 0$. If on the other hand term k indexes only 1 document, then its ability to discriminate among documents (this document from all others) is very high. These terms will receive the maximum weight: $n_k = 1$ and therefore $idf_k = log(N)$. We have applied this standard IDF weighting to the MeSH term dimensions and a modified IPFF (Inverse Pfam Frequency) weighting. Here each MeSH term is weighted by the log of the inverse number of Pfam families that contain proteins with referenced documents indexed by MeSH term k: $ipff_k = log(\frac{N^{PF}}{n_k^{PF}})$ where N^{PF} is the total number of Pfam families in the data set and n_k^{PF} is the number of Pfam families that contain a protein which reference a document indexed by MeSH term k. Both weightings improved recall by 20-40% for a given cosine similarity. IPFF weighting performed slightly better than IDF and all results reported were obtained with IPFF[9].

6.2.3 Algorithm for Pfam prediction in MeSH vector space

The classification algorithm employed here is closely related to the k-nearest neighbor algorithm [Duda et al., 2000]. Instead of considering the k nearest protein neighbors of a protein *i* to make a Pfam family prediction, our algorithm makes its prediction based on the proteins found in a fixed neighborhood of protein *i*. The neighborhood is defined by the cosine (see Eqn. 6.1). For a given protein *i* and neighborhood cosine $cos(\alpha)$, the neighborhood of protein *i* is delimited by a hyper-cone with an opening angle α and centered around the protein vector p_i. For each Pfam family the number of protein members within the hyper-cone neighborhood are counted. Our algorithm returns a ranking of Pfam families based on this number of proteins in the neighborhood, i.e. the family with most proteins in the neighborhood is ranked first. Note that not all Pfam families have the same number of protein members and our algorithm is biased towards predicting larger families, as they have more protein members. To weaken this effect, we limited the maximum family size to 20 protein members[10]. See Fig. 6.4 for an illustration.

6.3 Results

6.3.1 Pfam Predictions for Proteins

Figure 6.5 shows the prediction success of our algorithm in terms of proteins *recalled*, i.e. the number of proteins for which the Pfam family was predicted correctly. The x-axis indicates the cosine of the neighborhood angle. The y-axis on the left shows the number of proteins and the y-axis on the right the percentage of total proteins. Note that our algorithm does not necessarily make a prediction for a protein. The red, dashed curve shows for how many proteins predictions could be made at the respective neighborhood size. For small cosines (i.e. large angels α) predictions for all proteins can be made. But at $cos(\alpha) = 0.6$ ($\alpha \sim 53^o$), for example, only 47% of the proteins have a Pfam family prediction (only proteins can be predicted that have other proteins in the respective neighborhood).

[9] Besides IDF, IPF and IPFF weightings, we also applied entropy based weighting measures for MeSH term occurrences with proteins and Pfam families. Such entropy based measures take not only into account if a MeSH term occur rs with a protein or Pfam family, but also the frequency of co-occurrence (i.e. the number of documents indexed by the MeSH term and also referenced by the protein or Pfam family). No significant improvement in the results over IPF and IPFF was found.

[10] We also ranked Pfam families based on the number of protein members in the neighborhood normalized by Pfam family size. This biased the prediction towards smaller families. As there are many more small families in the data set, the overall prediction success of the algorithm was lower.

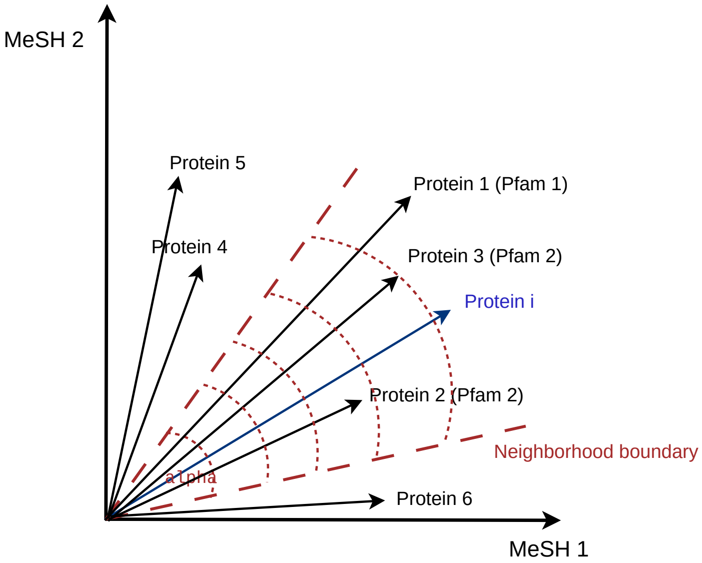


Figure 6.4: Illustration of classification algorithm and protein vectors in (reduced, two-dimensional) MeSH space. If protein i's Pfam family is to be predicted, the protein members of each Pfam family in the cosine neighborhood of protein *i* are counted. The Pfam families are then ranked by the number of members they have in the neighborhood. In the illustrated case, protein *i* has two protein members of Pfam family 2 and one from Pfam family 1 in its designated $cos(\alpha)$ neighborhood.

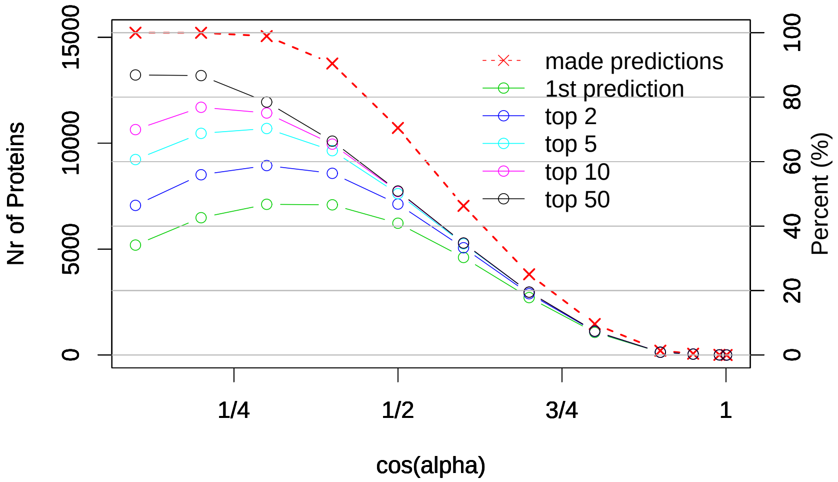


Figure 6.5: Prediction success for 15,217 proteins from 1611 Pfam families. At $cos(\alpha) = 0.3$ 47% of the proteins have their Pfam family predicted correctly by the first predicted family. For 70% of the proteins the correct Pfam family is ranked among the first 5 families. For 77% (an additional 7%) among the first 10 families.

The green curve in Fig. 6.5 shows the number of proteins for which the first ranked Pfam family was the correct family. At $cos(\alpha) = 0.3$ ($\alpha \sim 73^o$), for 47% (7115) of the proteins the first ranked family was the correct family. Note that the prediction is made into 1611 Pfam families. Disregarding knowledge of the family sizes, we would expect a success rate of $1/1611 = 0.06\%$ when predicting Pfam families by chance. Our prediction result represents a > 750 fold increase over such a Pfam prediction by chance. Of further interest is that for an additional 12% of the proteins the Pfam family ranked second is the correct family. For 70% of the proteins (an additional 11%) the correct Pfam family is among the Pfam families ranked first to fifth, and for 77% the correct family is ranked first to tenth.

6.3.2 “Misclassifications” into related families

These results raised the question if the top ranked Pfam families are very different and, in cases where the correct Pfam family is not ranked top, choosing the top ranked family would be a completely wrong classification, or, if the top ranked families are closely related, and therefore classification into the top ranked family would still provide the correct functional annotation of the protein in most cases. We identified the kinds of misclassifications between Pfam families that were most frequently made by our algorithm. The graph in Figure 6.6 shows the most frequent

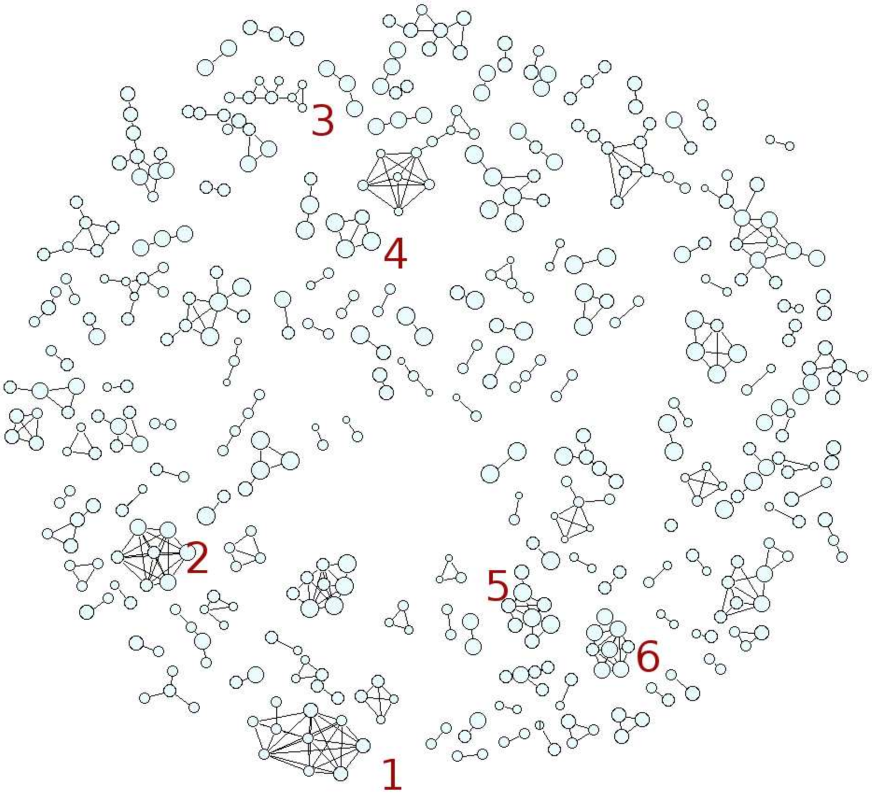


Figure 6.6: Misclassifications among Pfam families. The nodes in this directed graph are the Pfam families. The size of each node indicates the number of correctly predicted proteins for the respective family. The numbers 1-6 indicate the cliques of Pfam families discussed in more detail below. The graph was created with the Fruchterman-Reingold algorithm in Pajek [Batgelj and Mrvar, 2004].

misclassifications between Pfam families for the proteins that had their correct Pfam family ranked between second and tenth[11]. The cosine neighborhood was set at $cos(\alpha) = 0.4$ ($\alpha = 66^o$). The graph shows a directed link between two Pfam families if 3 or more proteins from the Pfam family where the link originates had the Pfam family where the link ends ranked higher than the correct Pfam family (i.e. the Pfam family where the link originated). Fig. 6.6 only shows the 473 Pfam families with such links among them[12]. The graph clearly indicates that Pfam families cluster into cliques in MeSH space. Many of the cliques are small, and there are fewer, more highly connected cliques that are larger.

We found three major types of Pfam families that were connected in cliques and we present

[11] Note that there are no mispredictions for the 47% of proteins that have their family ranked first.

[12] Many of the smaller families were filtered out by the cutoff of 3 proteins. We have observed, however, that the cliques shown are robust to parameter variations, such as protein cutoff or cosine neighborhood size.

PF00979: Reovirus outer capsid protein, Sigma 3

PF01518: Sigma NS protein

PF03084: Reoviral Sigma1/Sigma2 family

PF01616: Orbivirus NS3

PF01718: Orbivirus non-structural protein NS1, or hydrophobic tubular protein

PF01700: Orbivirus VP3 (T2) protein

PF01516: Orbivirus helicase VP6

PF00897: Orbivirus inner capsid protein VP7

PF00898: Orbivirus outer capsid protein VP2

PF00901: Orbivirus outer capsid protein VP5

Table 6.1: This clique contains 7 orbivirus protein families (5 structural and 2 non-structural) that are all linked to each other. In addition there are 3 Reovirus protein families. Orbiviruses are a genus under the RNA virus family Reoviridae, so they are related. See also the lower plot in Fig. 6.7, showing that MeSH terms Bluetongue virus and Reoviridae link the families in this clique. The Bluetongue virus is a species of the Orbivirus genus. It seems the Bluetongue virus is the most studied species of the Orbivirus genus.

2 examples for each type: (i) Pfam families related to different viruses, examples being the Orbiviruses of clique 1 and Rotaviruses in clique 2 in Fig. 6.6 and in Tables 6.1 and 6.2, (ii) Pfam families that are linked due to related enzymatic functions, examples being Hydrolases in clique 3 and Dehydrogenases in clique 4 in Fig. 6.6 and in Tables 6.3 and 6.4 and (iii) Pfam families that are subunits of proteins or protein complexes, examples being the ATP Synthase subunits in clique 5 and Cytochrome C Oxidase subunits in clique 6 in Fig. 6.6 and in Tables 6.5 and 6.6[13]. The different cliques of Pfam families are listed and discussed in the Tables 6.1 - 6.6 and their captions.

Plots of the protein-MeSH tf*ipff association matrix are shown in Figures 6.7. The upper plot shows that only a few MeSH terms are highly associated and specific to a clique and its Pfam families. The lower plot shows the 20 MeSH terms most associated with the proteins from the 6 cliques of Pfam families. These MeSH terms are very specific to the respective Pfam families. The indexing of the publications in MEDLINE must be very consistent and specific to achieve such high prediction success with such few, very specific MeSH terms. Considering that 85% of proteins only cite one publication, almost all the literature cited by correctly predicted proteins of a family needs to be indexed by this few, specific MeSH terms for the family.

[13] Note that especially the distinction between cliques based on enzymatic function and being a protein subunit is not always clearcut.

PF00426:	Outer Capsid protein VP4 (Hemagglutinin)
PF00434:	Glycoprotein VP7
PF00989:	Rotavirus major capsid protein VP6
PF00981:	Rotavirus RNA-binding Protein 53 (NS53)
PF01452:	Rotavirus non structural protein
PF01525:	Rotavirus NS26
PF02509:	Rotavirus non-structural protein 35

Table 6.2: The Pfam database lists all these 7 families as belonging to the Rotavirus genus, which like the Orbivirus genus in Table 6.1 belongs to the family of Reoviridae. The MeSH vocabulary and the indexing process of the respective literature were specific enough to separate these 2 cliques of protein families. They do share MeSH terms, for example ones related to the capsid proteins (see lower plot in Fig. 6.7).

PF02289:	Cyclohydrolase (MCH)
PF00795:	Carbon-nitrogen hydrolase
PF01425:	Amidase
PF00561:	alpha/beta hydrolase fold
PF01546:	Peptidase family M20/M25/M40
PF00557:	metallopeptidase family M24
PF01244:	Membrane dipeptidase (Peptidase family M19)
PF03575:	Peptidase family S51

Table 6.3: A clique of pfam families that are hydrolases. The relationship of these families is again captured by the MeSH terms. The three most associate MeSH terms with this clique are all located under D08.811.277-Hydrolases (see plot in Fig. 6.7). Note that there is not one MeSH term linking all Pfam families in this clique. The clique is also less highly connected and more chain-like than the highly connected cliques of virus families and protein subunits. This is a property we found for other enzyme related cliques as well.

PF00106:	short chain dehydrogenase
PF00107:	Zinc-binding dehydrogenase
PF00465:	Iron-containing alcohol dehydrogenase
PF00180:	Isocitrate/isopropylmalate dehydrogenase

Table 6.4: Four Pfam families of different dehydrogenases.

PF01991: ATP synthase (E/31 kDa) subunit

PF00231: ATP synthase

PF00213: ATP synthase delta (OSCP) subunit

PF04627: Mitochondrial ATP synthase epsilon chain

PF00119: ATP synthase A chain

PF00137: ATP synthase subunit C

PF01990: ATP synthase (F/14-kDa) subunit

PF01496: V-type ATPase 116kDa subunit family

Table 6.5: All families in this highly connected clique are ATP Synthase related. The main MeSH terms associated and linking the proteins of these Pfam families are Proton Translocating ATPases and Adenosinetriphosphatase. MeSH terms that proteins in this clique share with the proteins in the clique of Cytochrome C Oxidase in Table 6.6 are related to mitochondria. Both protein complexes are essential components of the cellular respiration pathway in mitochondria. Both cliques link up and form one larger clique when weaker links are shown in Figure 6.6.

PF00510: Cytochrome c oxidase subunit III

PF00015: Cytochrome C and Quinol oxidase polypeptide I

PF02936: Cytochrome c oxidase subunit IV

PF02285: Cytochrome oxidase c subunit VIII

PF02284: Cytochrome c oxidase subunit Va

PF01215: Cytochrome c oxidase subunit Vb

PF02238: Cytochrome c oxidase subunit VIIa

PF02046: Cytochrome c oxidase subunit VIa

Table 6.6: Clique of Cytochrome C Oxidase subunits. Like ATP Synthase (Tab 6.5), Cytochrome C Oxidase is an essential complex in the respiratory pathway of mitochondria. MeSH terms Cytochrome C Oxidase and Electron Transport Complex IV are clearly highly relevant and specific to these families (see lower plot in Fig. 6.7).

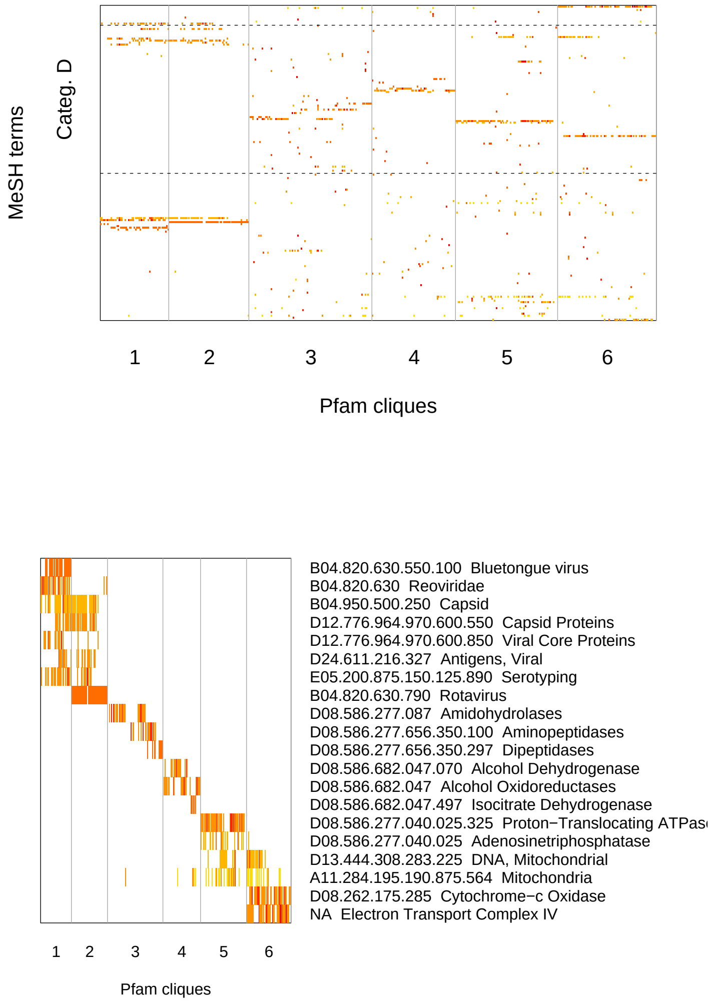


Figure 6.7: Image plots of the protein-MeSH tf*ipff association matrix for the 6 cliques of 45 Pfam families and a total of 477 proteins. The proteins of the respective Pfam families and cliques are the columns of the matrix and the MeSH terms are the rows. The upper image shows the 160 most significant MeSH terms, with at least one tf*ipff value of 10 or more with a protein. Colors indicate the magnitude of the matrix values, darker red indicating larger values, lighter yellow indicating smaller values. Category D MeSH terms, containing most of the protein and enzyme related MeSH terms, are indicated by the dashed horizontal lines. The lower image plot only shows the 20 most significant MeSH terms and what they are.

6.3.3 Factors influencing prediction success

We identified two factors that influence prediction success: (i) the number of protein members a Pfam family has (we also refer to this as *family size*) and (ii) the number of publications that a protein references. As mentioned previously, our classification algorithm is expected to perform better for larger families. Larger families close to smaller families can lead to misclassifications of the proteins from the smaller families. The top figure in Fig. 6.8 shows the correlation of prediction success with family size[14]. As family size increases, so does prediction success. The first ranked Pfam family is the correct Pfam family for 70% of proteins from families of size 15 and close to 80% from families with sizes of 20 proteins. Proteins from families with size 3 or 4 are predicted with a success rate of only 25%, however. The bottom figure shows how the prediction success is correlated with the number of publications that a protein references. For proteins with only one publication reference (85% of the proteins), at best 58% of the proteins can be predicted correctly by the first ranked Pfam family. For proteins with 5 publication references recall increases to 76% and for 10 publication references recall reaches 83%. 4 proteins had 40 or more publications, all of these were predicted correctly.

6.4 Discussion and Conclusions

Our study shows that Pfam families do indeed cluster in MeSH space. We have shown this for a large data set of 1611 Pfam families, whereas previous studies mostly have shown the separation of few sets of functionally distinct groups of genes or proteins in some keyword space (e.g. [Andrade and Valencia, 1998,Masys et al., 2001]). It should be noted that this clustering of sequence families in MeSH is achieved through the literature, not in some keyword space that is associated with the proteins directly. Such keywords, like the SwissProt keywords used in the study of MacCallum et al. [MacCallum et al., 2000], might be assigned with the physical properties of proteins, like sequence similarity, in mind. Clustering of Pfam families in such a keyword space would be less surprising.

It should also be noted again that we filtered the literature aggressively, that we allowed each document to be referenced by only one protein. Therefore, our results were not obtained by having proteins and Pfam families somehow linked "artificially" by shared publications.

Also, our algorithm performed an unsupervised classification, the algorithm did not "learn" to classify the Pfam families by fitting some parameters[15]. The classification results of our algorithm

[14] For both plots in Fig. 6.8, the neighborhood angle was not fixed but the best prediction for each protein was selected. These prediction rates are the best that could be achieved with our algorithm if the neighborhood size would be allowed to vary, for example with location of protein in MeSH space.

[15] We therefore did not perform a cross-validation study.

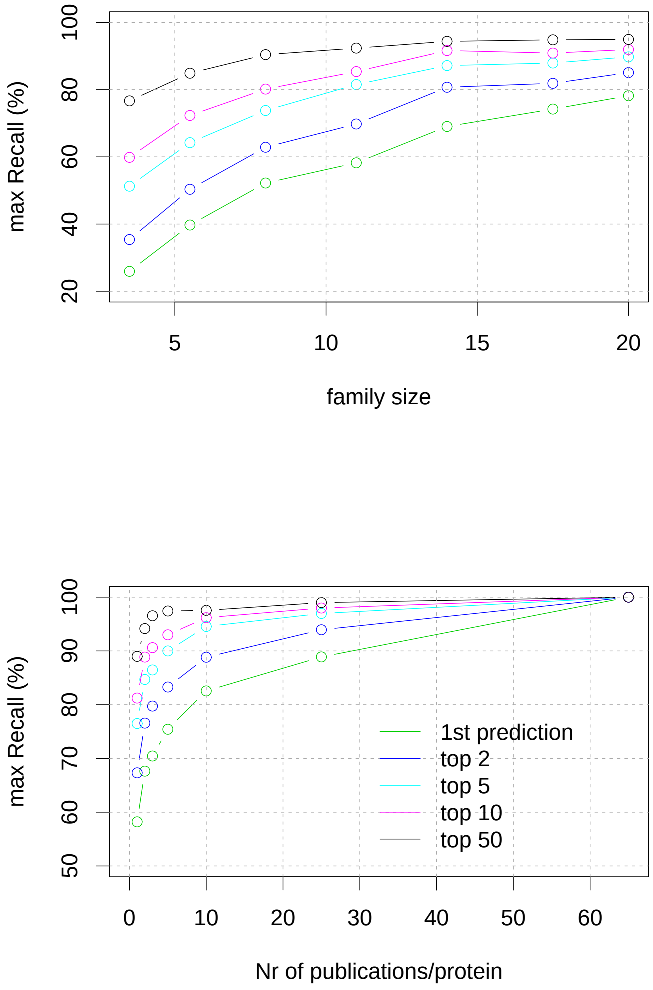


Figure 6.8: The dependence of Pfam prediction success on family size (top) and number of publications referenced per protein (bottom). Both factors are correlated with prediction success. Note that we found no correlation between family size and number of referenced publications, both factors influence prediction success independently of each other.

might, however, be improved by allowing the neighborhood cosine size to vary, for example with Pfam family size and/or location in MeSH space. This will require further study.

We have shown that Pfam families that are close in MeSH space, e.g. families that have many mispredictions of proteins among them, are related. Such “confusions” of functionally related Pfam families is for many Bioinformatics tasks actually desirable, as in many tasks, it is the functional class of a group of genes or proteins that is sought, not the specific Pfam family.

Our study showed that a few MeSH terms seem to determine and specify a Pfam family or a clique of related Pfam families. As mentioned before, successful prediction of proteins with few literature citations per protein requires highly consistent and specific indexing of publications by NLM in MEDLINE. It is surprising to us how well the indexing process must be performed to achieve our results. That only few, very specific MeSH terms specify a Pfam family probably explains why techniques like Latent Semantic Analysis (LSA) [Deerwester et al., 1990,Berry et al., 1995], which exploit synonymy and polysemy in a vocabulary, and which we tested for this task as well, did not improve our results significantly. Basically, Pfam families, or related cliques of Pfam families, are already orthogonal to each other in MeSH space, and there is little synonymy or polysemy in the vocabulary and indexing process.

We have shown that two factors greatly influence the ability to predict the Pfam families of proteins: the amount of literature that they have and the size of their family. The latter is partly an artifact of our classification algorithm. We want to explore adding more literature, first by filtering less aggressively and adding back some literature that is cited by proteins from more than one Pfam family. Such literature could be added with some weights, similar to the IPFF weighting of the MeSH terms. A second way to add literature would be by mining all of MEDLINE for the respective protein names and symbols. We have discussed some of the challenges that such an approach poses. However, we don’t know of a study comparing a body of literature and the specificity of its information when obtained from experts, like SwissProt citations, or when mined by entity identification. It would be interesting and informative to repeat our study with a body of literature obtained in the latter form, and compare it to the results obtained here.

Chapter 7

Summary and Possible Future Work

Work in two main areas of Bioinformatics was presented in this dissertation: gene expression analysis and automated mining of functional information from literature. SVD was presented as a well suited method for time series expression analysis, partly due to its robustness to noise in expression data and its ability to allow for easy visualization of the data. Two algorithms based on SVD were presented that identify significantly expressed genes and group the genes into co-expression clusters. In the second part of the work, a method to mine functional information from literature was presented. The usefulness of the developed method to the analysis of expression data was illustrated. A large scale validation study of the method was performed: the classification of proteins into sequence families based on literature was compared to the known, true sequence classification.

There are several ways in which the presented methods could be extended. Our expression analysis work focused on SVD, where the identified modes are linear combinations of the gene expression vectors. As expression data become more complex (e.g. longer time courses and more complicated processes that are observed) methods that can detect non-linear relationships among expression patterns might become more valuable[1]. Methods that might be useful are Non-linear PCA [Jolliffe, 1986, Schölkopf et al., 1996] or Kernel methods [Schölkopf and Smola, 2002], for example.

The literature mining method that was presented can be extended as well. We focused in the presented work on the MeSH vocabulary. The presented vector space model could be implemented with different vocabularies. Terms could, for example, be extracted from the literature directly[2]. We found in our work that publications in MEDLINE are indexed with very specific MeSH terms, e.g. the precise enzymatic function (EC class) an enzyme performs. This can cause two enzymes

[1] As was shown, for the current data sets, most times two linear modes suffice to “explain” the data.

[2] One problem with that approach is that most publications are not available as free text. MEDLINE only provides the abstract for publications.

that have related function, but not exactly the same function and therefore not exactly the same MeSH terms, to appear unrelated in MeSH term space (i.e. the protein vectors are orthogonal to each other in MeSH term space). The hierarchical organization of the MeSH vocabulary might be used to eliminate this problem. For example, if a publication, referenced by a protein or gene, is indexed with some MeSH term in MEDLINE, all MeSH terms above this MeSH term in the hierarchy could be added to the term vector of the document (or protein/gene). This approach is somewhat related to what is know as “spreading activation” [Salton and Buckley, 1988] (e.g. “spreading” the “indexing activation” up the term hierarchy). We explored this approach when classifying proteins into Pfam sequence families. Our classification results did not improve and worsened if we propagated the indexing activation to the top level of the MeSH hierarchy. We suspect that because many MeSH terms have multiple parent terms in the hierarchy, the activation spreads too fast and information is lost. If the propagation up the hierarchy could be constrained, maybe to a select set of branches in the MeSH tree, the approach might still be valuable. Further research might be done in this area.

Bibliography

[Affymetrix, 1999] Affymetrix (1999). *Gene Chip Analysis Suite User Guide*. Affymetrix, Santa Clara, CA.

[Affymetrix, 2005] Affymetrix (2005). Affymetrix. http://www.affymetrix.com.

[Alter et al., 2000] Alter, O., Brown, P., and Botstein, D. (2000). Singular Value Decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA*, 97(18):10101–10106.

[Andrade and Valencia, 1998] Andrade, M. and Valencia, A. (1998). Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. *Bioinformatics*, 14(7):600–607.

[Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., and et al. (2000). Gene ontology: tool for the unification of biology. (The Gene Ontology Consortium). *Nat Genet*, 25(1):25–29.

[Baeza-Yates et al., 1999] Baeza-Yates, R., Ribiero-Neto, B., and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Pearson Education.

[Bairoch et al., 2005] Bairoch, A., Apweiler, R., Wu, C., Barker, W., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M., Natale, D., O'donovan, C., Redaschi, N., and Yeh, L. (2005). The Universal Protein Resource (UniProt). *Nucleic Acids Research*, pages D154–D159.

[Baldi and Long, 2001] Baldi, P. and Long, A. (2001). A bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics*, 17(6):509–519.

[Bateman et al., 2004] Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L. L., Studholme, D. J., Yeats, C., and Eddy, S. R. (2004). The Pfam Protein Families Database. *Nucleic Acids Research*, 32:D138–D141.

[Batgelj and Mrvar, 2004] Batgelj, V. and Mrvar, A. (2004). Pajek - Program for Large Network Analysis. http://vlado.fmf.uni-lj.si/pub/networks/pajek/.

[Benson et al., 2004] Benson, D., Karsch-Mizrachi, I., Lipman, D., Ostell, J., and Wheeler, D. (2004). Genbank: update. *Nucleic Acids Res*, 32:D23–6.

[Berry, 1992] Berry, M. (1992). Large-scale sparse singular value computations. *International Journal of Supercomputer Applications*, 6:13–49.

[Berry et al., 1993] Berry, M., Do, T., Obrien, G., Krishna, V., and Varadhan, S. (1993). Svdpackc: Version 1.0 user's guide. Technical report, Knoxville: University of Tennessee.

[Berry et al., 1995] Berry, M., Dumais, S., and O'Brien, G. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595.

[Borg and Groenen, 1997] Borg, I. and Groenen, P. (1997). *Modern Multidimensional Scaling: Theory and Applications*. Springer Verlag, New York.

[Brown et al., 2000] Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Ares, M. J., and Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, 97(1):262–7.

[Browne et al., 2001] Browne, E., Wing, B., Coleman, D., and Shenk, T. (2001). Altered cellular mRNA levels in human cytomegalovirus-infected fibroblasts: Viral block to the accumulation of antiviral mrnas. *Journal of Virology*, 75(24):12319–30.

[Cattell, 1966] Cattell, R. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, 1:245–76.

[Challacombe et al., 2004] Challacombe, J., Rechtsteiner, A., Gottardo, R., Rocha, L., Brown, E., Shenk, T., Altherr, M., and Brettin, T. (2004). Evaluation of the host transcriptional response to human cytomegalovirus infection. *Physiological Genomics.*, 18(1):51–62.

[Chen et al., 1996] Chen, L., Hodgson, K., and Doniach, S. (1996). A lysozyme folding intermediate revealed by solution x-ray scattering. *J Mol Biol*, 261:658–71.

[Cho et al., 1998] Cho, R. J., Campbell, J. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockart, D. J., and Davis, R. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1):65–73.

[Chu et al., 1998] Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P., and I., H. (1998). The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705.

[Consortium, 2001] Consortium, I. H. G. S. (2001). Initial sequencing and analysis of the human genome. *Nature*, 409.

[Cover and Thomas, 1991] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience.

[Deerwester et al., 1990] Deerwester, S., Dumais, S., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

[Deprette, 1988] Deprette, F., editor (1988). *SVD and signal processing, Algorithms, Applications and Architectures.* North-Holland.

[DeRisi et al., 2000] DeRisi, J., Iyer, V., and Brown, P. O. (2000). The mguide: A complete guide to building your own microarrayer. *http://cmgm.stanford.edu/pbrown/mguide/*.

[DeRisi Lab, 2005] DeRisi Lab (2005). Microarrays.org. http://www.microarrays.org.

[D'haeseleer, 2000] D'haeseleer, P. (2000). *Reconstructing Gene Networks from Large Scale Gene Expression Data.* PhD thesis, University of New Mexico.

[D'haeseleer et al., 2000] D'haeseleer, P., Liang, L., and Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726.

[Duda et al., 2000] Duda, R., Hart, P., and Stork, D. (2000). *Pattern Classification*. Wiley, New York, NY, 2nd edition.

[Duggan et al., 1999] Duggan, D., Bittner, M., Chen, Y., Meltzer, P., and Trent, J. (1999). Expression profiling using cDNA microarrays. *Nature Genetics*, 21:10–14.

[Dumais, 1990] Dumais, S. (1990). Enhancing performance in latent semantic indexing.

[Eisen and Brown, 1999] Eisen, M. and Brown, P. (1999). DNA arrays for analysis of gene expression. *Methods Enzymol.*, 303:179–205.

[Eisen et al., 1998] Eisen, M., Spellman, P., Brown, P., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95:14863–14868.

[Everitt and Dunn, 2001] Everitt, B. S. and Dunn, G. (2001). *Applied Multivariate Data Analysis*. Edward Arnold.

[Everitt et al., 2001] Everitt, S., Landau, S., and Leese, M. (2001). *Cluster Analysis*. Edward Arnold, 4th edition.

[Fesel and Coutinho, 1998] Fesel, C. and Coutinho, A. (1998). Dynamics of serum igm autoreactive repertoires following immunization: strain specificity, inheritance and association with autoimmune disease susceptibility. *Eur. J. Immunol.*, 28(11):3616–29.

[Fodor et al., 1993] Fodor, S. P. A., Rava, R. P., Huang, X. C., Pease, A. C., Holmes, C. P., and Adams, C. L. (1993). Multiplexed biochemical assays with biological chips. *Nature*, 364:555–556.

[Fraley and Raftery, 1998] Fraley, C. and Raftery, A. (1998). MCLUST: Software for model-based cluster and discriminant analysis. Technical Report 342, Department of Statistics, University of Washington.

[Friedman and Tukey, 1974] Friedman, J. and Tukey, J. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 23:881–89.

[Funk and Reid, 1983] Funk, M. and Reid, C. (1983). Indexing consistency in MEDLINE. *Bull Med Libr Assoc*, 71(2):176–83.

[Garcia, 1992] Garcia, A. (1992). Large-amplitude nonlinear motions in proteins. *Phys Rev Lett*, 68.

[Golub and Van Loan, 1996] Golub, G. and Van Loan, C. (1996). *Matrix Computations*. Johns Hopkins University Press, 3rd edition.

[Golub et al., 1999] Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caliguiri, M., Bloomfield, C., and Lander, E. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–7.

[Hackstadt and Mallavia, 1982] Hackstadt, T. and Mallavia, L. (1982). Sodium and potassium transport in herpes simplex virus-infected cells. *Journal of General Virology*, 60:199–207.

[Hardy and RM., 1994] Hardy, W. and RM., S.-G. (1994). Herpes simplex virus inhibits host cell splicing, and regulatory protein icp27 is required for this effect. *J Virol.*, 68(12).

[Harris et al., 2004] Harris, M., Clark, J., Ireland, A., Lomax, J., Ashburner, M., and et al. (The Gene Ontology Consortium) (2004). The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res*, 32:D258–D261.

[Hastie et al., 2000] Hastie, T., Tibshirani, R., Eisen, M., Alizadeh, A., Levy, R., Staudt, L., Chan, W., Botstein, D., and Brown, P. (2000). 'Gene Shaving' as a method for identifying distinct sets of genes with similar expression pattern. *GenomeBiology.com*, 1.

[Holter et al., 2001] Holter, N., Maritan, A., Cieplak, M., Fedoroff, N., and J.R., B. (2001). Dynamic modeling of gene expression data. *Proc Natl Acad Sci USA*, 98:1693–98.

[Holter et al., 2000] Holter, N., Mitra, M., Maritan, A., Cieplak, M., Banavar, J., and Fedoroff, N. (2000). Fundamental patterns underlying gene expression profiles: simplicity from complexity. *Proc. Natl. Acad. Sci. USA*, 97(15):8409–8414.

[Hughes et al., 2000] Hughes, T., Marton, M., Jones, A., Roberts, C., Stoughton, R., Armour, C., Bennett, H., Coffey, E., Dai, H., He, Y., Kidd, M., King, A., Meyer, M., Slade, D., Lum, P., Stepaniants, S., Shoemaker, D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M., and Friend, S. (2000). Functional discovery via a compendium of expression profiles. *Cell*, 102:109–26.

[Hyvarinen, 1999] Hyvarinen, A. (1999). Survey on Independent Component Analysis. *Neural Computing Surveys*, 2:94–128.

[Ideker et al., 2001] Ideker, T., Galitski, T., and Hood, L. (2001). A new approach to decoding life: Systems Biology. *Annu. Rev. Genomics Hum. Genet.*, 2:343–372.

[Iyer et al., 1999] Iyer, V. R., Eisen, M. B., Ross, D. T., Schuler, G., Moore, T., Lee, J. C. F., Trent, J. M., Staudt, L. M., Hudson, J., J., Boguski, M. S., and et al. (1999). The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87.

[Jain and Dubes, 1988] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ.

[Jenssen et al., 2001] Jenssen, T., Laegreid, A., Komorowski, J., and Hovig, E. (2001). A literature network of human genes for high-throughput analysis of gene expression. *Nat. Genet.*, 28(1):21–28.

[Jessup and Sorensen, 1994] Jessup, E. and Sorensen, D. (1994). A parallel algorithm for computing the singular-value decomposition of a matrix. *Siam Journal on Matrix Analysis and Applications*, 15:530–48.

[Jolliffe, 1986] Jolliffe, I. (1986). *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag.

[Kanehisa, 2000] Kanehisa, M. (2000). *Post-Genome Informatics*. Oxford University Press.

[Kanehisa Lab., 2005] Kanehisa Lab. (2005). KEGG: Kyoto Encyclopedia of Genes and Genomes. http://www.genome.ad.jp/kegg/.

[Kanji, 1993] Kanji, G. K. (1993). *100 Statistical Tests*. Sage.

[Kitano, 2001] Kitano, H., editor (2001). *Foundations of systems biology*. MIT Press, Cambridge, Mass.

[Knudsen, 2002] Knudsen, S. (2002). *A Biologist's Guide to Analysis of DNA Microarray Data*. John Wiley & Sons, New York.

[Kohonen, 1995] Kohonen, T. (1995). *Self Organizing Maps*. Springer-Verlag, Berlin.

[Kohonen, 2001] Kohonen, T. (2001). *Self Organizing Maps*. Springer-Verlag, Berlin.

[Lander, 1996] Lander, E. S. (1996). The new Genomics: Global views of biology. *Science*, 274:536–539.

[Lander, 1999] Lander, E. S. (1999). Array of hope. *Nature Genetics*, 21(3-4).

[Leming, 2002] Leming, S. (2002). www.gene-chips.com. http://www.gene-chips.com.

[Li and Wong, 2001a] Li, C. and Wong, W. (2001a). Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc Natl Acad Sci USA*, 98:31–36.

[Li and Wong, 2001b] Li, C. and Wong, W. (2001b). Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biology*, 2:1–11.

[Liebermeister, 2002] Liebermeister, W. (2002). Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18:51–60.

[Lukashin and Fuchs, 2001] Lukashin, A. V. and Fuchs, R. (2001). Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics*, 17(5):405–414.

[MacCallum et al., 2000] MacCallum, R., Kelley, L., and Sternberg, M. (2000). SAWTED: structure assignment with text description–enhanced detection of remote homologues with automated SWISS-PROT annotation comparisons. *Bioinformatics*, 16(2):125–129.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observation. In Le Cam, L. and Nyeman, J., editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability.*, volume 1. University of California Press.

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

[Masys et al., 2001] Masys, D., Welsh, J., Lynn Fink, J., Gribskov, M., Klacansky, I., and Corbeil, J. (2001). Use of keyword hierarchies to interpret gene expression patterns. *Bioinformatics*, 17(4):319–26.

[National Library of Medicine, 2004] National Library of Medicine (2004). MeSH Fact Sheet. http://www.nlm.nih.gov/pubs/factsheets/mesh.html.

[National Library of Medicine, 2005] National Library of Medicine (2005). Pubmed. http://www.ncbi.nlm.nih.gov/entrez/.

[NCBI, 2004] NCBI (2004). GenBank. http://www.ncbi.nlm.nih.gov/Genbank/index.html.

[Park et al., 1993] Park, H., Davidson, D., Raaka, B., and Samuels, H. (1993). The herpes simplex virus thymidine kinase gene promoter contains a novel thyroid hormone response element. *Molecular Endocrinology*, 7:319–330.

[Press et al., 1992] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipes in C.* Cambridge University Press, Cambridge, 2nd edition.

[R Development Core Team, 2004] R Development Core Team (2004). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. 3-900051-07-0.

[Raychaudhuri et al., 2000] Raychaudhuri, S., Stuart, J. M., and Altman, R. B. (2000). Principal components analysis to summarize microarray ex-periments: application to sporulation time series. In *Pacific Symposium on Biocomputing*, pages 452–463.

[Rechtsteiner et al., 2003] Rechtsteiner, A., Gottardo, R., Rocha, L., and Wall, M. (2003). Singular Value Decomposition for analysis of gene expression. In Spang, R., Beziat, P., and Vingron, M., editors, *Currents in Computational Molecular Biology. Proceedings of the The Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB 2003)*, pages 275–276, Berlin.

[Rechtsteiner and Rocha, 2004a] Rechtsteiner, A. and Rocha, L. (2004a). MeSH key terms for validation and annotation of gene expression clusters. In Gramada, A. and Bourne, P. E., editors, *Currents in Computational Molecular Biology. Proceedings of the Eight Annual International Conference on Research in Computational Molecular Biology (RECOMB 2004)*, pages 212–213.

[Rechtsteiner and Rocha, 2004b] Rechtsteiner, A. and Rocha, L. (2004b). Use of MEDLINE's MeSH terms for automated functional annotation. In *ISCB Rocky Mountain Regional Bioinformatics Meeting*.

[Rechtsteiner et al., 2005] Rechtsteiner, A., Strauss, C., and Rocha, L. (2005). Clustering of Pfam in MeSH space. Technical report, Los Alamos National Laboratory Internal Report.

[Richards, 1993] Richards, J. (1993). *Remote Sensing Digital Image Analysis.* Springer-Verlag.

[Rijsbergen, 1979] Rijsbergen, C. (1979). Information Retrieval (WWW edition). http://www.dcs.gla.ac.uk/Keith/Preface.html.

[Romo et al., 1995] Romo, T., Clarage, J., Sorensen, D., and Phillips, G. (1995). Automatic identification of discrete substates in proteins: singular value decomposition analysis of time-averaged crystallographic refinements. *Proteins*, 22:311–21.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). On the use of spreading activation methods in automatic information. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 147–160. ACM Press.

[Schena et al., 1995] Schena, M., Shalon, D., and Davis, R.W.and Brown, P. (1995). Quantitative monitoring of gene-expression patterns with a complementary-DNA microarray. *Science*, 270(5235):467–470.

[Schölkopf and Smola, 2002] Schölkopf, B. and Smola, A. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA.

[Schölkopf et al., 1996] Schölkopf, B., Smola, A., and Muller, K.-R. (1996). Nonlinear component analysis as a kernel eigenvalue problem. Technical report, Tuebingen: Max-Planck-Institut fur biologische Kybernetik.

[Shatkay et al., 2000] Shatkay, H., Edwards, S., Wilbur, W., and Boguski, M. (2000). Genes, themes and microarrays: using information retrieval for large-scale gene analysis. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 8:317–328.

[SIB/EBI, 2004] SIB/EBI (2004). UniProt/Swiss-Prot. http://www.ebi.ac.uk/swissprot/.

[Sonnhammer et al., 1997] Sonnhammer, E., Eddy, S., and Durbin, R. (1997). Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins*, 28:405–420.

[Spellman et al., 1998] Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9:3273–97.

[Stekel, 2003] Stekel, D. (2003). *Microarray Bioinformatics*. Cambridge University Press, Cambridge, UK.

[Strang, 1998] Strang, G. (1998). *Introduction to Linear Algebra*. Wellesley Cambridge Press, Wellesley, MA.

[Szallasi, 2001] Szallasi, Z. (2001). Genetic network analysis - from the bench to computers and back. In *2nd International Conference on Systems Biology*.

[Tamayo et al., 1999] Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E., and Golub, T. (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, 96:2907–2912.

[Tavazoie et al., 1999] Tavazoie, S., Hughes, J., Campbell, M., Cho, R., and Church, G. (1999). Systematic determination of genetic network architecture. *Nat Genet*, 22(3):281–5.

[Terry Speed’s Group, 2003] Terry Speed’s Group (2003). Always log spot intensities and ratios. *http://www.stat.berkeley.edu/users/terry/zarray/Html/log.html*.

[The Gene Ontology Consortium, 2004] The Gene Ontology Consortium (2004). The Gene Ontology Website. http://www.geneontology.org/.

[Troyanskaya et al., 2001] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17:520–25.

[Venter et al., 2001] Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., and et al. (2001). The sequence of the human genome. *Science*, 291(5507):1304–1351.

[Wall et al., 2001] Wall, M., Dyck, P., and Brettin, T. (2001). SVDMAN – Singular Value Decomposition analysis of microarray data. *Bioinformatics*, 17:566–568.

[Wall et al., 2003] Wall, M., Rechtsteiner, A., and Rocha, L. (2003). *A Practical Approach to Microarray Data Analysis*, chapter Singular Value Decomposition and Principal Component Analysis, pages 91–109. Kluwer Academic Publishers, Boston, MA.

[Yang et al., 2001] Yang, Y. H. amd Dudoit, S., Luu, P., and Speed, T. P. (2001). Normalization for cdna microarray data. In *SPIE BiOS 2001*, San Jose, California.

[Yeung et al., 2001a] Yeung, K., Fraley, C., Murua, A., Raftery, A., and Ruzzo, W. (2001a). Model-based clustering and data transformation for gene expression data. *Bioinformatics*, 17(10):977–987.

[Yeung and Ruzzo, 2001] Yeung, K. and Ruzzo, W. (2001). Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774.

[Yeung et al., 2001b] Yeung, K. Y., Haynor, D. R., and Ruzzo, W. L. (2001b). Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318.

[Yeung et al., 2002] Yeung, M., Tegner, J., and Collins, J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proc Natl Acad Sci USA*, 99:6163–68.

[Zweiger, 1999] Zweiger, G. (1999). Knowledge discovery in gene-expression-microarray data: mining the information output of the genome. *Trends in Biotechnology*, 17(11).

Glossary[3]

C

cDNA stands for ’complementary DNA’, a single stranded DNA molecule that is complementary to a full-length mRNA. Such strands are typically 500-5000 bases long.

cDNA microarrays microarray technology for transcript level measurements that use cDNA strands as probes on the chip.

cell-cycle the temporal cycle of cellular processes that leads to the division of a cell. Typically this cycle is divided into (i) G1 phase, growth and preparation of the chromosomes for replication, then (ii) S phase, synthesis of DNA (and centrosomes), then (iii) G2 phase, which is preparation for (iv) M phase, mitosis, when the actual division of the cell and nucleus occur.

central dogma (of molecular biology) the coding of genes in DNA which are transcribed into mRNA which in turn are translated into proteins.

co-expression genes that have similar expression values across different conditions or experiments are called *co-expressed* (’similar’ needs to be defined and measured with a similarity or distance measure). Clustering algorithms attempt to identify groups (or clusters) of co-expressed genes. Co-expressed genes are assumed to be co-regulated and functionally related.

complementary (sequences) are nucleic acid sequences that can form a double-stranded structure with another nucleic acid sequence by following base-pairing rules (adenine (A) pairs with thymine (T) and cytosine (C) with guanine (G)). The complementary sequence to GTAC for example, is CATG.

comparative hybridization (see ’competitive hybridization’)

[3] This is a still incomplete glossary.

competitive hybridization in cDNA microarrays the ’competitive’ hybridization process of reverse transcribed and fluorescently labeled cDNA strands from test and control mRNA samples. Competitive hybridization is supposed to take care of artificial ’spot-effects’ on microarray chips.

D

DNA (nucleotides) a nucleotide is an organic compound with a molecular structure that contains a nitrogen-containing unit (base) linked to a sugar and a phosphate group. Four different nucleotides (cytosine, thymine, adenine and guanine), differing only in their base, comprise the DNA.

E

eigengene we denote as eigengenes the right-singular vectors (row vectors of V^T) of a SVD of a *gene expression matrix* X. The eigengenes are linear combinations of the gene expression vectors (rows) of a *gene expression matrix* X. The first few eigengenes typically capture the dominant patterns of expression change contained in a *gene expression matrix*.

eigenassay we denote as eigenassays the left-singular vectors (column vectors of matrix U) of a SVD of a *gene expression matrix* X. The eigenassays are linear combinations of the expression profiles (columns) of a *gene expression matrix* X.

expression signature (or ’finger-print’) a characteristic state of the Transcriptome that distinguishes different cell, tissue or phenotypes. Such ’signatures’ could be very beneficial for early detection of diseases, for example.

expression vector a gene’s expression values across different samples/time points. Typically a row in a *gene expression matrix* X where the rows refer to the genes and the columns to the samples/time points.

expression profile the measured expression values of a all genes on a microarray. Typically a column in a *gene expression matrix* X where the rows refer to the genes and the columns to the samples/time points.

F

fold change an approach often chosen to identify genes significantly changing in expression between two experiments or to filter genes before further processing. A threshold on the ratio of a gene's expression over some control expression value is applied. Typical values for such fold-change thresholds range between 2 and 3 (genes with lower ratio than the threshold being filtered out).

functional theme we use this term to denote significant association of a group of genes with some biological function, e.g. a group of co-expressed genes that can be associated with some cellular process.

G

gene expression matrix a matrix of gene expression values obtained from microarray experiments. Typically a row refers to a gene's expression vector across the measured assays and a column represents the expression values of all the genes measured with a single assay/microarray.

genotype the genetic constitution of an individual, either overall or at a specific *locus* in the genome (see *phenotype*).

H

homolog (or homologous gene) - two or more genes whose sequences are significantly related because of close evolutionary relationship, either between species (*orthologs*) or within a species (*paralogs*).

high-throughput experiments/technologies

hybridization process in which two complementary nucleic acids strands interact through hydrogen bonds so that double stranded DNA-DNA or DNA-RNA structures are formed. Between DNA strands adenine (A) pairs with thymine (T) and cytosine (C) with guanine (G). For example, the complementary sequence to GTAC is CATG. The tendency of complementary nucleic acid strands to hybridize is the basis of microarray technology.

I

induction genes that increase in expression in time, or whose expression level is observed to be above their normal baseline level in a certain condition, are called *induced (*see also *repression).*

L

locus a unique chromosomal location defining the position of an individual gene or DNA sequence.

M

MeSH Medical Subject Heading nomenclature used by the National Library of Medicine to index all publications in MEDLINE. MeSH is a hierarchical nomenclature of 22,000 key terms organized under main headings like Organisms, Diseases, Chemicals and Drugs, etc. The MeSH key term hierarchy can be linked to genes (or other biological entities like proteins) through publications in MEDLINE that mention the respective genes.

N

Nucleotide is an organic compound with a molecular structure that contains a nitrogen-containing unit (base) linked to a sugar and a phosphate group. Four different nucleotides (cytosine [C], thymine [T], adenine [A] and guanine [G]), differing only in their base, encode the DNA sequence of genes.

O

oligo-nucleotides relatively short sequence of nucleotides, in the oligo-nucleotide chip technology on the order of 25 nucleotides.

P

phenotype the observable characteristics of a cell or organism (see *genotype*).

polysemy discussed in the context of Latent Semantic Analysis. Polysemy refers to the phenomenon that one keyword can refer to several concepts. (see also *synonymy*)

principal component score vectors (also just called ***scores*****)** the column vectors of the orthogonal matrix T of the principal component decomposition of a matrix X:$X = \mathbf{1}\bar{\mathbf{a}}^T + TP^T$. The scores are the coordinates of the row objects of X in the space of the *principal component loading vectors.*

principal component loading vectors (also just called *principal components)* the row vectors of the orthogonal matrix P^T of the principal component decomposition of a matrix X:$X = \mathbf{1}\bar{\mathbf{a}}^T + TP^T$. The principal components are the eigenvectors of the covariance matrix of X.

R

repression genes that decrease in expression in time, or whose expression level is observed to be below their normal baseline level in a certain condition, are called repressed (see also *induction*).

S

scree plot term introduced by Cattell [Cattell, 1966] to denote the plot of the singular values (or singular values squared) identified by SVD. Cattell proposed to use the scree plot to identify the significant components in a SVD analysis. The scree plot tends to decrease sharply initially, for the components that are associated with signals in the data, and then levels off for the components that are mostly associated with noise.

scores see *principal component score vectors*

spot effects artifical effects on measured expression values due to artifacts of the spotted probes on a cDNA microarray. Competitive hybridization of a reference and the test sample mRNA and subsequent normalization of the two measured fluorescent signals is supposed to eliminate spot effects.

synonymy discussed in the context of Latent Semantic Analysis. Synonymy refers to the phenomenon that several keywords can refer to the same concept. (see also *polysemy*)

T

transcriptional response see *expression vector.*

Transcriptome all the transcripts of an organism (similar to ’Genome’ and ’Proteome’)

V

vector space model in Information Retrieval, a model that represents documents as vectors in keyword or term space (the terms contained in the documents). A similarity measure between documents can be defined (typically the cosine of the angle between the document vectors), and documents are retrieved based on their similarity with the query term vector. See [Baeza-Yates et al., 1999, Deerwester et al., 1990, Berry et al., 1995].

Appendix A

Biology Background

A.1 The Central Dogma of Molecular Biology

The *central dogma* of molecular biology (see also Fig. A.1) states that the information for a protein's sequence which is encoded in deoxyribonucleic acid (DNA) by sequences of four different nucleotides, is first *transcribed* into *messenger RNA* (mRNA) and then *translated* into proteins. The number of genes in humans was originally thought to be around 100,000 but has been revised to between 25,000 and 30,000 [Consortium, 2001, Venter et al., 2001][1]. Proteins are sometimes referred to as the *molecular machines* of a living cell, they are involved in one way or another in most of the biological processes in a cell. The control of protein abundance in the cell is an important mechanism by which the cell controls its internal state and responds to external stimuli. Protein abundance is in large part controlled by regulation of transcription. It is this process, the regulation of transcription, that microarrays are aimed at measuring for whole genomes. Sometimes the complete set of transcripts in a cell is referred to as *Transcriptome,* similarly to the *Genome* for the set of all genes of an organism, or the *Proteome,* the set of all proteins. Microarrays allow us to measure the Transcriptome of a cell, or a set of cells.

Physicists could regard the mRNA transcript levels as a subset of the state variables describing the molecular state of a cell. Being able to determine and eventually model the internal states of a cell, versus only being able to observe external, phenotypical properties of a cell, promises, among other things, greater success in earlier and more specific diagnosis of diseases and a better understanding of life on the cellular and molecular level.

[1] The actual number of different proteins in a cell is actually higher. Due to *alternative splicing* of the mRNA transcripts before translation into proteins, more than one protein can be encoded by a single gene. *Post-translational modifications* further diversify the protein population of a cell.

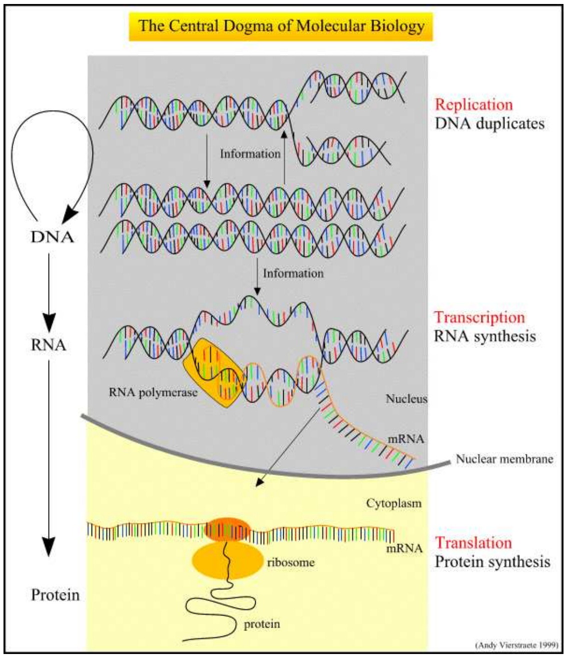


Figure A.1: The central Dogma of molecular biology. Figure adapted from http://www.accessexcellence.org.

Appendix B

Gene Expression Data Analysis

A gene expression data set usually contains data from more than one experiment, i.e. microarray, as multiple samples or time points are assayed for comparison. The data is therefore multidimensional and commonly organized in matrix form, with each row corresponding to a gene or mRNA transcript and each column to a sample or a time point. A specific value x_{ij} in the matrix corresponds to the measured expression value for a specific gene *i* for assay *j*, where the assay typically corresponds to a certain sample or time point. The row *i* in a gene expression matrix represents gene *i*'s expression values across the samples and is referred to as the *expression vector* or *transcriptional response* of gene *i*. Alternatively, the elements of column *j* of the expression matrix is referred to as the *expression profile* of the corresponding assay (see also section **??** and specifically section 2.1 for more detailed definitions).

B.1 Transformation, Normalization and Filtering

The purpose of transformation and normalization of data is to identify and remove artificial sources of variation. A frequently performed transformation on microarray gene expression data is to take the logarithm of the data. For data from cDNA microarrays experiments, upregulation and downregulation by a certain factor have the same absolute value after log transformation, just opposite signs. Further, it has been shown that log transformed data have a more normal distribution than raw intensities or ratios alone [Terry Speed's Group, 2003], which can simplify further statistical analysis. Another observation that has been made is that the variance of the expression values is less dependent on their mean after log transformation, indicating that the error causing processes are multiplicative rather than additive [Terry Speed's Group, 2003]. After log transformation the error processes become additive and their effects are independent of the absolute magnitude of expression.

A so-called *global normalization* of all data on a chip is performed to account for differences in labeling efficiency between the two fluorescent dyes in cDNA microarrays. A constant adjustment is used to force the distribution of the log-expression values to have a median of zero on each slide[1]. After such *global normalization*, or *global scaling*, of the data, often times a form of *local standardization* of the individual gene expression vectors is performed. The most frequent standardizations are centering of the gene expression vectors, so the means are zero, and scaling so the standard deviation or variance is one.

Filtering involves reducing the data by removing uninformative genes whose expression levels did not change or were below a user-defined threshold. Filtering of genes is most often performed by removing genes based on a *fold-change* criterion or based on the variance across samples or time points. The fold-change filter removes genes whose expression change across the samples is lower than a pre-specified fold-change with respect to a reference expression value (which could be the average expression value of a gene across the samples or an expression measurement taken before the start of an experiment, e.g. before virus infection). Such filters have to be applied before the standardization of variance.

B.2 Distance Measures

To be able to explore the similarity of gene transcriptional profiles in expression space, first a *similarity* or *distance measure* needs to be defined. The kind of distance measure used can impact the output of further analysis and is therefore important to consider. The measures that have mostly been used to analyze gene expression data have been the Euclidean Distance and the Pearson correlation[2]. For time series data the Pearson correlation has been suggested as the often more appropriate measure [Knudsen, 2002] as a similar pattern of expression change among genes is more indicative of similarity than similar amplitude, or magnitude of expression. However, if the common standardizations of mean centering and unit variance are applied to the gene transcriptional profiles, the two measures are closely related. The Pearson Correlation between two variables or vectors $\mathbf{x} = \{x_1, ..., x_n\}$ and $\mathbf{y} = \{y_1, ..., y_n\}$ is defined as

$$r_{xy} = \frac{1}{n}\sum_{i=1}^{n}(\frac{x_i - \overline{\mathbf{x}}}{\sigma_x})(\frac{y_i - \overline{\mathbf{y}}}{\sigma_y}) \tag{B.2.1}$$

With the means $\overline{\mathbf{x}} = \overline{\mathbf{y}} = 0$ and the standard deviations $\sigma_x = \sigma_y = 1$, it follows that

[1] More complicated normalization techniques have been suggested for cases where dye biases can depend on spot overall intensity and location on the array (print-tip effects) [Yang et al., 2001].

[2] Note that the Pearson correlation is actually a similarity measure, with $r_{xy} = -1$ indicating largest dissimilarity between vectors x and y and $r_{xy} = 1$ indicating largest similarity. Any similarity measure can be transformed into a distance measure, for example a Pearson distance can be defined as $d^P_{xy} = 1 - r_{xy}$.

$$r_{xy} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i$$

The square of the Euclidean Distance for the two vectors is defined as

$$\begin{aligned} d_{xy} &= \sum_{i=1}^{n}(x_i - y_i)^2 \\ &= \sum_{i=1}^{n}(x_i^2 + y_i^2 - 2x_i y_i) \end{aligned} \tag{B.2.2}$$

and using $\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 = \sigma_y^2 = \frac{1}{n}\sum_{i=1}^{n} y_i^2 = 1$ it follows that

$$\begin{aligned} d_{xy} &= 2n(1 - \frac{1}{n}\sum_{i=1}^{n} x_i y_i) \\ &= 2n(1 - r_{xy}) \end{aligned} \tag{B.2.3}$$

More complex measures, for example based on Information Theoretic measures like *Mutual Information* [Cover and Thomas, 1991], might detect more complex causal relationships than the linear Euclidean distance and Pearson correlation.

In the analyses presented in the dissertation, the gene expression data have been log transformed, mean centered and standardized to unit variance.

B.3 Levels of Analysis of Gene Expression Data

Gene expression data analysis can be organized into three levels of increasing complexity:

1. The “simplest” analysis is on the level of single genes, where one seeks to find if a single gene in isolation is differently expressed in different conditions.

2. The second level is a multi-variate analysis of gene expression data from multiple conditions or time points. The goal of such an analysis can be to group samples that have been assayed based on their gene expression profiles or to identify groups of genes responding in a similar way in a time series expression experiment. The most frequently applied methods for this kind of analysis are clustering algorithms.

3. At the third level the goal is to infer the underlying gene (and protein) networks that are ultimately responsible for the patterns observed. Most of the work here attempts to re-engineer the gene regulatory networks from the expression data.

The work here is mostly concerned with level 2. For completeness we give a brief overview of the other levels of analysis as well.

B.3.1 Level 1: Detecting Differentially Expressed Genes

To identify genes that are differently expressed in two samples of mRNA one approach commonly used is a simple *fold change* approach. A gene is declared to have significantly changed if the absolute factor of change between the expression levels in the two samples is larger than a certain threshold. Typical threshold values range from 2 to 3 [Szallasi, 2001, Cho et al., 1998, Browne et al., 2001]. It has been argued, however, that such a simple fold change criterion is unlikely to yield optimal results [Szallasi, 2001]. First, because the variance in measured expression levels depends on the mean expression level, a specific fold-change can have different significance in different regions of the spectrum of expression levels (though the log-transform can take care of some of that effect [Terry Speed's Group, 2003, Szallasi, 2001]). Second, artificial random effects, like stochasticity in reverse transcription, especially for rare transcripts, can cause fluctuations in measurements of transcript abundance that do not reflect real differences [Szallasi, 2001]. It has been reported that a 1.5 - 2 fold 'pseudo-change' in 1% of the quantified gene population can be expected by chance in microarray experiments [Szallasi, 2001]. If one notes first the compound effect of usually evaluating multiple experiments, secondly, that thousands of genes are assayed on a single chip, and thirdly, that for most experiments one doesn't expect a biologically caused change in expression for more than 10% of the genes, this is a significantly large number of genes.

If replicate measurements are available, a more sophisticated approach to the question of differential expression is the use of a *t-test* [Baldi and Long, 2001]. In a t-test, the empirical means and variances of the replicates of two conditions i and j are used to compute a normalized distance between the two populations in the form

$$t = (\overline{m}_i - \overline{m}_j) \Big/ \sqrt{\frac{\sigma_i^2}{n_i} - \frac{\sigma_j^2}{n_j}} \tag{B.3.4}$$

where, for each population, $\overline{m}$ and σ are the estimates for the mean and standard deviation and n is the number of replicates in the two conditions. t follows approximately a student distribution. When t exceeds a certain threshold depending on the confidence level selected, the two populations of replicate measurements are considered to be different. Because in the t-test the distance between the population means is normalized by the empirical standard deviations, this has the potential for addressing some of the shortcomings of the fold-change criterion. There is a fundamental problem with the t-test for microarray data, however. For many experiments there are no replicates, or that number is small because experiments remain costly and tedious to repeat.

B.3.2 Level 2: Multivariate Analysis of Gene Expression Data

Multi-variate analysis of gene expression data has typically focused on the application of clustering techniques. Genes can be clustered based on their transcriptional response vectors (the vector of expression values across conditions). Such genes are referred to as co-expressed and the assumption is that these genes also might be co-regulated and therefore functionally related. Samples can also be clustered based on their expression profiles. The goal here often is to find a grouping of the samples based on their expression profiles, for example into samples that are from healthy and samples that are from non-healthy tissue.

Clustering is a fundamental technique in exploratory data analysis and pattern discovery. The application of clustering algorithms to some data assumes the preexistence of groupings of the objects to be clustered. Random noise and artifacts may have obscured these groupings. The objective of the clustering algorithm is to recover the original grouping of the data. Sometimes clustering algorithms are divided into *supervised* and *unsupervised* algorithms. Most often though, supervised clustering algorithms are referred to as *classification* algorithms. In classification tasks information about the groupings of some of the data needs to be present. Typically a set of reference vectors or classes is given and at least some of the objects (the *training data*) are assigned to one (or multiple) of these. (Unsupervised) Clustering typically tries to infer groupings from the structure of the data directly, when no predefined set of vectors or classes are known. Gene expression analysis so far has mostly been exploratory. The functions of many genes under different conditions are still unknown and therefore (unsupervised) clustering methods are more commonly used for gene expression analysis[3]. Clustering can also be seen as providing a reduction of the dimensionality of the expression data. If a few clusters of co-expressed genes can be identified in a gene expression data set and these clusters can be associated with cellular functions or processes, then the data set with potentially thousands of assayed genes has been “reduced” to a few significant underlying cellular processes that generated the data.

[3] Some attempts with classification methods, such as support vector machines, have been made [Brown et al., 2000].

B.3.3 Level 3: Gene Regulatory Model Inferences from Gene Expression Data

A goal in Functional Genomics is to be able to build detailed gene regulatory models. Attempts have been made to use gene expression data to infer and reverse engineer gene regulatory networks. However, the task of reverse engineering of gene regulation networks from time series data has proven difficult at the current stage of gene expression assaying. The number of time points assayed is typically in the lower teens, the number of genes involved are in the thousands and the data is noisy. Some examples and reviews on the sate of gene regulation inference from expression data are given in [D'haeseleer et al., 2000, D'haeseleer, 2000, **?**, Szallasi, 2001].

B.4 Clustering Algorithms applied to Gene Expression Data

Hierarchical Clustering Eisen et al. [Eisen et al., 1998] popularized the application of the *agglomerative hierarchical clustering algorithm* to gene expression data. The Pearson Correlation was used as a similarity metric. The algorithm starts with N clusters containing a single gene each. At each step in the iteration the two closest clusters are merged into a larger cluster. In the average-linkage version of the algorithm, distance between two clusters is defined as the distance between the averages of the clusters' gene expression vectors. After N-1 steps, all the genes have been organized into one hierarchy of clusters and this hierarchy can be visualized in a tree where the branch length corresponds to the measured distance of the corresponding nodes (i.e. clusters).

Two variants of the average-linkage algorithm are single-linkage and complete-linkage algorithms. The iterative agglomeration of the two closest clusters remains the same, but distance between clusters is defined differently. For single-linkage, the distance between two clusters is defined as the distance of the closest pair of elements from the two clusters. Complete linkage defines the distance of two clusters as the distance of the pair of elements from the two clusters with the largest distance.

These three clustering techniques will in many cases produce different partitionings of data, as each favors, or is biased towards, a different cluster topology. Complete linkage favors spherical clusters, whereas single-linkage is able to detect lower-dimensional clusters which can be extended in only some dimensions of the space. Average linkage is placed somewhere between the two but in general also favors spherical clusters. The bias towards spherical clusters can be a problem for data where the different dimensions, i.e. assays, are correlated, as is usually the case for time series data and other expression data where several samples come from the same categories, i.e. healthy and diseased samples.

K-means Tavazoie et al. [Tavazoie et al., 1999] applied a K-means clustering algorithm [MacQueen, 1967] to a yeast cellcycle gene expression data set [Cho et al., 1998]. Tavazoie et al. chose the Euclidean distance as the distance measure. The K-means algorithm partitions the N genes into K clusters, where K has to be pre-set by the user. K initial *cluster centroids* are chosen - usually at random - and each gene is then assigned to the cluster with the nearest centroid. Next, the centroid for each cluster is recalculated by averaging all gene expression vectors belonging to the respective cluster. This process is iterated until no more changes occur in the partitioning, or the amount of change falls below a pre-defined threshold. K-means clustering minimizes the sum of the squared distances of all genes to their respective cluster centroids. Different random initial seeds can be tried to assess the robustness of the clustering results.

Downsides to K-means clustering are that the number of clusters to be detected is an input to the algorithm and needs to be known. For example, Tavazoie chose K=30 clusters for the yeast cellcycle data. The same data has been clustered into 5 coexpression clusters by visual inspection by Cho et al. [Cho et al., 1998]. Further, like the average-linkage and complete-linkage hierarchical clustering methods, K-means favors spherical cluster topologies. As argued previously, such an assumption can in general not be made for gene expression data.

Multi-Variate Gaussians Mixture Models Yeung et al. [Yeung et al., 2001a] introduced an algorithm that fits multi-variate Gaussian distributions to gene expression data[4]. An Expectation-Maximization (EM) algorithm is used to maximize the likelihood and fit the multi-variate Gaussian mixture models. Rather than classifying each gene into one specific cluster, membership is indicated by the distributions' values for each of the Gaussian distributions. This can be interpreted as allowing each gene to have a *fuzzy membership degree* in more than one cluster, i.e. distribution. Such a feature might be valuable for gene expression data as genes can participate in more than one biological process.

Because the models can be flexible, allowing for many different covariance matrices for the Gaussian distributions, these models encompass other clustering schemes, like the Fuzzy K-means algorithm (as the name suggests, the fuzzy version of the 'crisp' K-means algorithm discussed above). However, more complex implementations of the algorithm can also fit unique and quite varying covariance matrices for each cluster. Different covariance matrices that allow for clusters with different sizes, (elliptical) shapes and orientations in expression space can be fit. Therefore, unlike K-means or hierarchical clustering, they are less biased towards specific cluster topologies. All of this comes at a cost, of course, of having to fit

[4] This algorithm is also known as a variant of the Fuzzy K-Means, or Z-means algorithm.

increasing numbers of parameters the more complex the models become. For the most complex model, allowing size, shape and orientation to vary, there are $n+n(n+1)/2$ parameters to estimate for each cluster, where n is the dimensionality of the data [Yeung et al., 2001a].

Yeung et al. [Yeung et al., 2001a] propose to also use the probabilistic nature of the models to estimate the correct number of clusters, i.e. number of Gaussian distributions. They provide an implementation of the algorithm that uses the Bayesian Information Criterion (BIC) to help select appropriate number of clusters as well as model classes, i.e. covariance matrices [Fraley and Raftery, 1998].

Self-Organizing Map The Self-Organized Map (SOM) [Kohonen, 1995] algorithm belongs to the class of Artificial Neural Networks (ANN). It has been applied to microarray gene expression data of the yeast cell cycle as well as a study of hematopoietic differentiation of four human cell lines [Tamayo et al., 1999]. The cluster centers in a SOM are typically located on a grid. A SOM performs a neighbor-preserving projection of the data from their higher dimensional space onto the grid-space, which typically is 1, 2 or 3-dimensional. At each iteration, a randomly selected gene expression vector is chosen and it then 'attracts' the nearest cluster center, plus some of its neighbors in the grid. Over time, fewer cluster centers are updated at each iteration, until finally only the nearest cluster is drawn towards each gene, placing the cluster centers in the center of gravity of the surrounding gene expression vectors.

Like in K-means, the user has to specify the number of clusters and therefore have some a priori knowledge about the number of clusters to expect. In addition, the grid topology, including the dimensions of the grid and the number of nodes in each dimension need to be specified. For example, 8 clusters could be mapped to a 2x4 2D grid or a 2x2x2 3D cube. The different geometries will impose different structures on the data.

Of benefit of the grid structure is that it helps to visualize the results. Nearby nodes in the grid will correspond to clusters with more similar expression patterns than clusters corresponding to nodes further away in the grid.

The following general statement about clustering in [Jain and Dubes, 1988] also has to be considered for clustering of gene expression data:

> There is no single best criterion for obtaining a partition [clustering] because no precise and workable definition of *cluster* exists. Clusters can be of any arbitrary shapes and sizes in a multidimensional pattern space. Each clustering criterion imposes a certain structure on the data, and if the data happens to conform to the requirements of a particular criterion, the true clusters are recovered.