Analysis of Molecular Expression Patterns and Integration with Other Knowledge Bases using Probabilistic Bayesian Network Models

E.J. Moler (ejmoler@lbl.gov), I.S. Mian (Smian@lbl.gov)

Mailstop 74-197, Department of Cell and Molecular Biology, Life Sciences Division
Lawrence Berkeley National Laboratory, Berkeley, CA 94720
Final Draft, March 2000

Abstract

How can molecular expression experiments be interpreted with $>10^4$ measurements per chip? How can one get the most quantitative information possible from the experimental data with good confidence? These are important questions whose solutions require an interdisciplinary combination of molecular and cellular biology, computer science, statistics, and complex systems analysis.

The explosion of data from microarray techniques present the problem of interpreting the experiments. The availability of large-scale knowledge bases provide the opportunity to maximize the information extracted from these experiments. We have developed new methods of discovering biological function, metabolic pathways, and regulatory networks from these data and knowledge bases. These techniques are applicable to analyses for biomedical engineering, clinical, and fundamental cell and molecular biology studies.

Our approach uses probabilistic, computational methods that give quantitative interpretations of data in a biological context. We have selected Bayesian statistical models with graphical network representations as a framework for our methods. As a first step, we use a naïve Bayesian classifier to identify statistically significant patterns in gene expression data. We have developed methods which allow us to a) characterize which genes or experiments distinguish each class from the others, b) cross-index the resulting classes with other databases to asses biological meaning of the classes, and c) display a gross overview of cellular dynamics. We have developed a number of visualization tools to convey the results. We report here our methods of classification and our first attempts at integrating the data and other knowledge bases together with new visualization tools.

We demonstrate the utility of these methods and tools by analysis of a series of yeast cDNA microarray data and to a set of cancerous/normal sample data from colon cancer patients. We discuss extending our methods to inferring biological pathways and networks using more complex dynamic Bayesian networks.

Table of Contents

Acknowledgements

This work was supported by the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098

LBNL Technical report number LBNL-44826.

Introduction

New techniques for interpreting molecular expression experiments are needed in the face of massively parallel acquisition of these patterns with techniques such as cDNA microarrays*(Jan 1999)*. The quantities of data preclude complete analysis by inspection. We are pursuing new methods of analysis which will utilize the full value of these data and knowledge bases for biological and biomedical research. Our ultimate goal is to discover new biological pathways, networks, and regulatory mechanisms, to identify new molecular expression mechanisms leading to cancer and other developmental phenomena, and to build predictive models of biological systems.

We have several concerns which are guiding our selection and development of methods.

Criteria for Selecting and Developing Methods

Automate pattern identification in the data due to the large volume of data
Integrate heterogeneous information types to infer interactions in complex systems
Support hypothesis testing and inference using quantitative statistical measures
Treat many-to-many mappings between genes and pathways/networks
Integrate hierarchical levels of information and views of function
Model the underlying stochasticity of biological processes
Utilize learning algorithms that account for missing information in a principled way
Require extensibility of the methods from coarse-level characterization to more intricate interactions

We have developed a road map of goals with roughly increasing sophistication of analysis. We have invented and selected methods of analysis which provide a development path with increasing sophistication, building on the first proof-of-principle steps, and yielding useful discoveries and predictions at each phase.

Clustering/Classification	discovering patterns within the data, generating statistical descriptions of expression patterns, quantifying distinguishing characteristics of the patterns
Integration	find patterns which signify function, regulation, etc. which are known from other knowledge bases, combine classification results with other discriminative methods (support vector machines, etc)
Inference	discover new category members and network links using classification and analysis of integrated knowledge
Model Extension	classification methods of a)continuous data with discrete variables, and b)including more complex conditional dependencies in the statistical model between elements and across time
Hypotheses Testing	supporting the domain expert (biologists and biomedical researchers), provide likelihood of hypothesized relationships and factors given the integrated data and knowledge bases
Reverse Engineering	find network and pathway links, look for causality, build predictive models

We have implemented significant portions of the first three phases in a working prototype of the analysis. Investigations into the 'model extension' phase and beyond are ongoing. This report focuses on the first three phases of development.

Background

Microarray technology for 'omics' studies is a rapidly expanding field. An excellent review of microarray technology at the beginning of 1999 is the special issue of 'Nature Genetics' *(Supplement Jan 1999)*. The applications include fundamental studies of eukaryotic cellular

dynamics *(DeRisi 1997; Chu 1998; Eisen 1998; Spellman 1998; Spellman 1998)*, and signature expression patterns of cancerous tissues *(Alon 1999; Perou 1999)*. There are several published methods noted below which approach the problem of extracting meaningful patterns of co-expression from large-scale microarray data. These methods do not meet all of the criteria which we have identified. Some methods of clustering and classification are complementary to probabilistic networks in their capabilities and are potentially very useful in combination.

Graphical models and probabilistic networks have been the subject of research in the machine learning community for many years. The literature on these subjects is voluminous. For excellent reviews, see references *(Heckerman 1997; Jensen 1998)*. Graphical models allow the representation of complex statistical relationships in an intuitive way. They also provide one framework for describing the statistical operations and assumptions used in analysis. Probabilistic networks are statistical descriptions of data and hidden variables. The structure and parameters of the network can be *learned* by a variety of methods. Bayesian networks are probabilistic networks whose structure and parameters are improved by an algorithm which starts with a prior network and uses the new data. Using Bayesian networks, it is possible to incorporate domain-specific expert knowledge into the analysis in a principled manner in the prior network. One can also use minimal information priors on the network to learn patterns and structure *a priori*.

Existing Methods of Expression Array Analysis

The published methods of clustering gene expression patterns to date include heirarchical clustering *(Eisen 1998)*, Fourier analysis for cell-cycle time series *(Spellman 1998)*, k-means *(Tavazoie 1999)* and self-organised maps *(Golub 1999; Tamayo 1999)*. While useful for an inspection analysis of the data, these methods do not provide a framework for a full statistical analysis or constrained extensions in complexity of the description of the data. Some methods are well suited to emphasizing particular features in the data of a carefully constructed series of experiments, for example the Fourier analysis for cell-cycle regulated genes. A more rigorous and generalized statistical treatment will yield rich interpretations of the data.

Most integration and inference efforts to date have involved visual inspection of the clusters, e.g. to identify genes which change during sporulation of yeast*(Chu 1998)* or to find similarity by tissue type*(Alon 1999; Golub 1999; Perou 1999)*. Tavazoie and co-workers used their clusters as a basis for discovering upstream promoter sequences *(Tavazoie 1999)*, as did Spellman, et al*(Spellman 1998)*.

There are published papers addressing the problem of inferring network architecture using boolean networks*(Liang 1998)*, coupled continuous non-linear differential equations*(Chen 1999)*, and neural networks*(Weaver 1999)*. We propose using a generalized statistical framework which allows a maximally rich, quantitative interpretation of the data and carries measures of uncertainty and partial certainty.

Our Approach: Graphical Models + Probabilistic Networks

Graph theory provides a unifying framework for encoding statistical relationships and operations. It is the union of statistics and computer science. The discovery and evaluation of complex statistical relationships can be implemented using *machine learning* techniques, automatically identifying statistically significant features among collections of heterogeneous

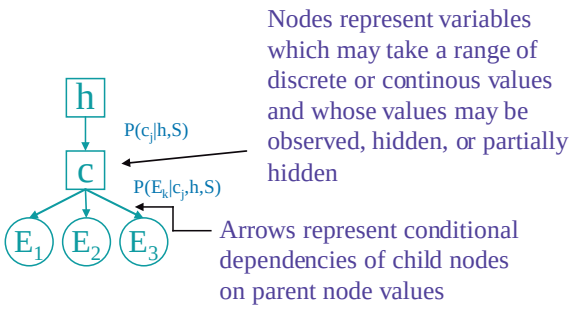
types of information. See the excellent reviews *(Heckerman 1997; Jensen 1998)* for a more complete description of Bayesian networks and associated techniques.

The Value of Probabilistic Networks

The theory of probabilistic networks is used to automate pattern identification in data, provide quantitative measures of cross-correlation, assess complex hypotheses and decisions against known data, and reverse engineer network structure all within the same framework while accounting for missing and noisy data in a principled way. This allows the application of increasingly sophisticated analyses.

The Value of Graphical Model Representations

Graphical models are intuitive visual representations of statistical relationships encoded by probabilistic networks. The nodes represent variables and the arrows represent influences between variable values. More specifically, a node represents a probability distribution of the variable value and the arrows specify possible conditional dependancies. The lack of an arrow signifies conditional independence: two variables not connected by an arrow are specified to be independent given the values of all common parent nodes. In other words, with no arrow, there is no direct influence of one variable on the other, but there may be indirect influence through an another factor.The graphical model gives only a qualitative view of the relationships. The quantitative aspects are determined by the parameters of the network model. One can think of each node as having a table of functionals which in some way combine the inputs to produce the output probability of the variable. This conditional probability table specifies a node-variable's dependence on all possible combinations of the parent variables' distributions. The dimensionality of the conditional probability matrix, or transition matrix, is determined by the number of arrows coming into the node.



A Simple Graphical Model

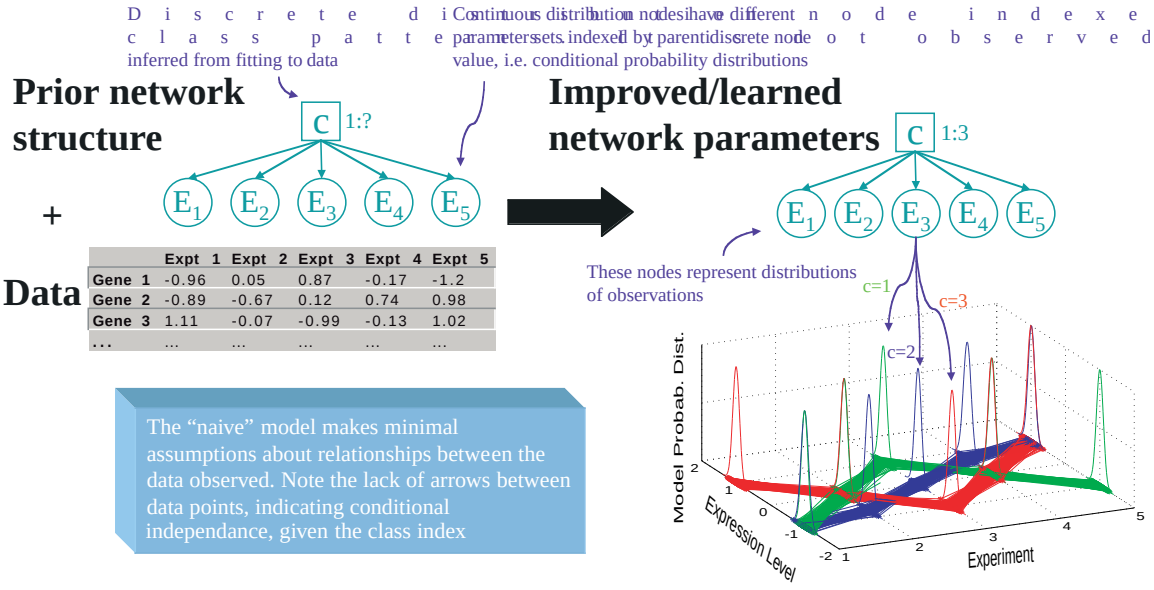
These models, representing networks of statistical relationships, provide a framework for compactly communicating statistical assumptions, operations, and results of inferences. This becomes particularly important when more complex relationships are treated. For example, some node-variables can represent molecular concentrations measured from array-expression data, other nodes can represent common promoter sites, and the arrows can represent in control of specific gene transcription through these sites.

General Approach

The classifier searches for a statistical model that best predicts the data. More specifically, we search for the maximum *a postori* (MAP) model h given the data E and the space of all possible models S using the form,

$$P(h \mid E,S) = \frac{P(E \mid h,S)P(h \mid S)}{P(E,S)} \qquad \textbf{Bayes' Rule}$$

Using Bayes' rule, models m
i.e. the likelihood of the data given the mo



Learning Network Parame

Given a network structure, w
parameters using the data. S
the methods.

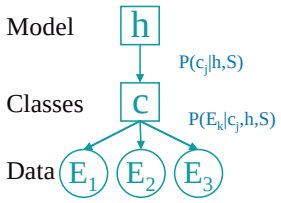
Naïve Bayes classifier

Our first model is a Finite
because it assumes very few
graphical
what those patterns should b
integrated with (
of a set of classes that defi

optimum number of classes and parameters for the pdfs are learned using a combination of a Monte Carlo search through the model space and an Expectation Maximization (EM) search for locally optimum parameters. The simplest search mode does not search for covariance between statistical parameters or attributes. This method of finding classes of attribute (expression) patterns in the data can be interpreted as a graph-theoretical tree of probabilistic relationships. The graph-theory approach to Bayesian networks provide a path to expanding the complexity of the learned relationships to covariance and beyond. See the Appendix and the references therein for details of the mathematical development.

Naïve Bayes Graphical Model

The graphical model of the naïve Bayes classifier is simple. The model contains one variable representing a class-index (c) and the others representing observed data (E). In this network, the class-index is discrete and enumerates the classes. The data nodes can be either discrete or continuous. The discrete data nodes contain tables of conditional probabilities, one set for each value of the class-index. Continuous data nodes contain tables of conditional probability distributions. For example, using a multivariate normal distribution, there is one pair of mean and standard deviation parameters for each of the discrete parent's index.



The Naïve Bayes Classifier

The Autoclass Implementation of the Naïve Bayes Classifier

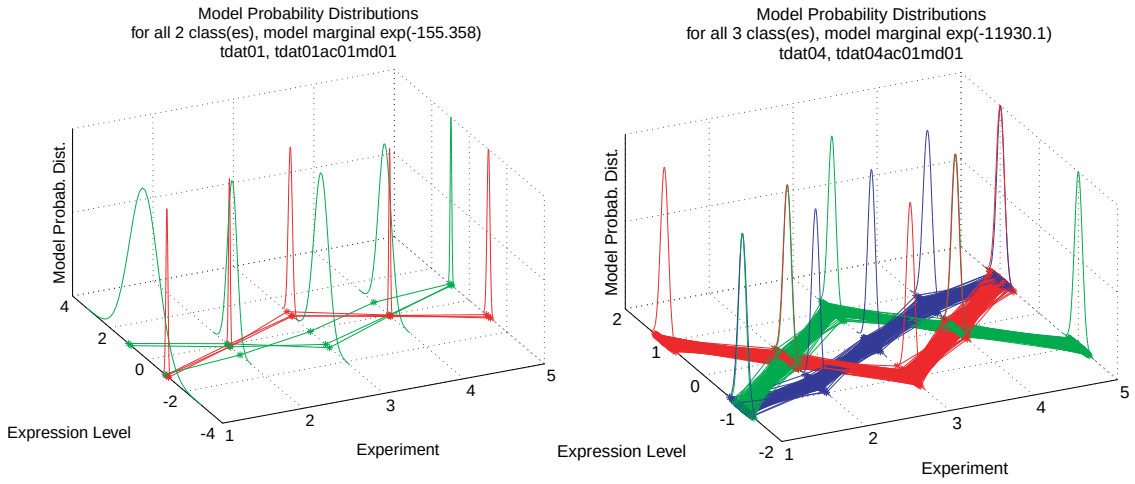
The actual model which autoclass implements is slightly more complex because of the way it treats missing data: every 'data' attribute has a discrete variable input which has the states 'observed' or 'not observed' to handle missing data. Other methods exist for evaluating partially observed network states when looking for optimum parameters or when learning node-value probabilities given a set of observations. AUTOCLASS also implements block-covariance and supports discrete, enumerated data nodes. We have not used these features in the initial applications.

The Toy Dataset - illustrating the method

Data is made up for illustration with 3 distinct patterns and random additive noise. The artificial data is composed of 3 different patterns with 300,200,and 100 respective cases generated from the sum of three different template patterns and an additive random noise function. The classification results are real. The software algorithm identified the best statistical model of the data to have 3 classes with the correct probability distributions. Each class defines a probability distribution across each attribute (experiment) and is indexed by the variable C. Data are plotted

with color matching the highest class membership probability. The learned class probability distribution functions are plotted in the 3rd dimension. Colors indicate class for each distribution.

One feature of the Bayesian network is the use of *priors*: probabilities which are assigned before taking into account the new observations. The priors explicitly describe the assumptions and expectations used in the models. The naïve Bayes classifier assumes minimal information. All models are equally likely *a priori*, regardless of the number of classes. Also, all classes in a model are equally likely. This prior builds in penalties for increased complexity because introducing an additional class reduces the prior likelihood of every class. This mechanism which prevents over fitting of the data also accounts for the number of observations, i.e. the amount of evidence. With fewer observations it is likely that there will be fewer classes because the strength of evidence for a new class must outweigh the cost of reducing the likelihood of all other classes. The figure demonstrates the results of a classification run where there are a total of 600 observations vs. a run with only 6 observations. Under-classification can occur if there are few examples of a class present in the data.

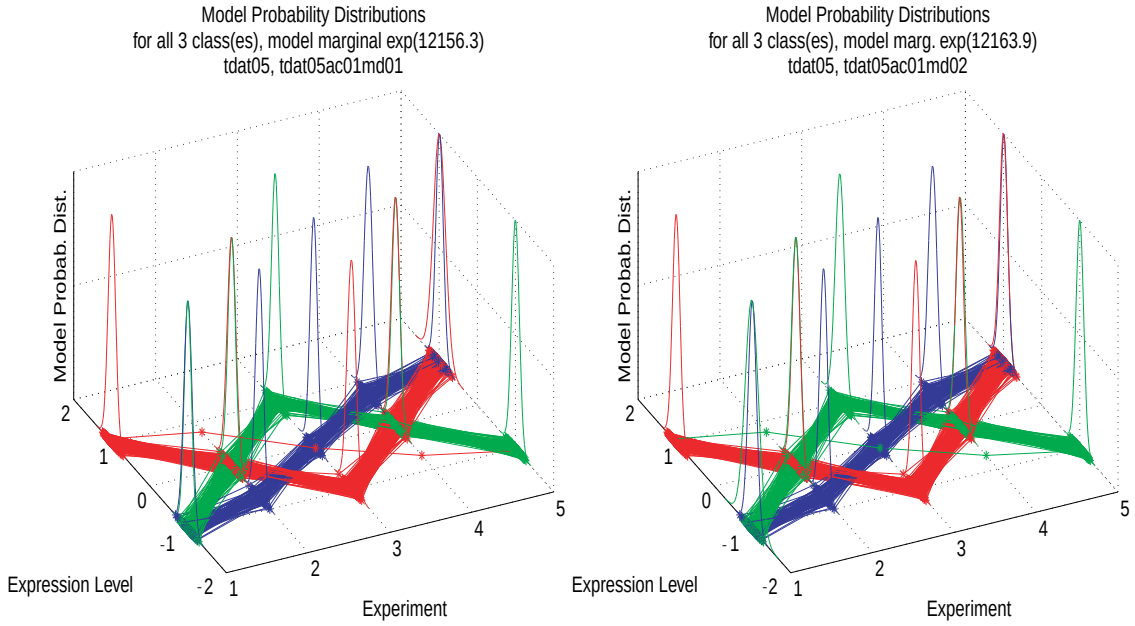


Two Models Learned from Small and Large Datasets

Fewer samples result in fewer classes being identified, more samples begin to outweigh prior estimates and assumptions

One useful feature for interpreting the model results is the marginal likelihood of the model given the data. This is an evaluation of the global description of the data by the model and determines a goodness of fit. The usefulness of the model likelihood becomes apparent when comparing the best identified models. If the models are sufficiently similar in their effectiveness

in describing the statistics of the data, their marginal likelihoods will be similar. In the case of insufficient evidence, several models may be found equally likely. This situation thus easily identified.



Two models with Essentially Equal Likelihoods

The model marginal likelihood is a measure of goodness of fit to the data. The single trace in the graph of expression levels falls mostly into different classes in the two models.

Methods, Testing, and Results

Classifications and analysis of the data were performed for two systems: yeast cell cycle series and cancerous/normal human colon tissues.

Applications to Yeast Gene Expression Analysis

Clustering and Classification: Finding Patterns and Distinguishing Features in the Data

The published differential gene expression data for the budding yeast *Saccharomyces Cerevisiae(DeRisi 1997; Chu 1998; Eisen 1998; Spellman 1998; Spellman 1998)* provides relative mRNA concentrations for each of ~6000 open reading frames (ORFs) across 78 experiments. These data are from genome-wide expression studies of the cell-division-cycle, sporulation, the diauxic shift, and mutant strains.

The differential expression data in the yeast experiments were reduced in the following way: the raw intensity data were corrected for background and then were reduced to a normalized form

$$I = \frac{I_{cy5} - I_{cy3}}{I_{cy5} + I_{cy3}}$$

This form has the same advantages as the log-ratio method enumerated by Eisen, but also has the additional features that 1) it minimizes the errors associated with background subtraction from low intensity signals, and 2) it constrains the expression levels to a domain of -1 to +1. To reduce complication in this first analysis, genes for which there were more than 10% missing measurements (>7) were removed from the data set. The resulting data set contains 5687 genes with 2846 bad or 'missing' data points. In contrast to other methods, no pre-selection of the data based on expression levels was necessary, nor did we shift or rescale the expression patterns.

Attribute #	Experiment	Attribute #	Experiment	Attribute #	Experiment
1	Clb2_2	27	cdc15_080	53	elu150
2	Cln3_2	28	cdc15_090	54	elu180
3	Cln3_1	29	cdc15_100	55	elu210
4	gal+-	30	cdc15_110	56	elu240
5	Alpha000	31	cdc15_120	57	elu270
6	Alpha007	32	cdc15_120	58	elu300
7	Alpha014	33	cdc15_130	59	elu330
8	Alpha021	34	cdc15_140	60	elu360
9	Alpha028	35	cdc15_150	61	elu390
10	Alpha035	36	cdc15_160	62	spo00
11	Alpha042	37	cdc15_160	63	spo005
12	Alpha049	38	cdc15_170	64	spo020
13	Alpha056	39	cdc15_180	65	spo050
14	Alpha063	40	cdc15_190	66	spo070
15	Alpha070	41	cdc15_200	67	spo090
16	Alpha077	42	cdc15_210	68	spo115
17	Alpha084	43	cdc15_220	69	spo_ndt80
18	Alpha091	44	cdc15_240	70	spo_delete_early
19	Alpha098	45	cdc15_250	71	spo_delete_mid
20	Alpha105	46	cdc15_270	72	diaux1
21	Alpha112	47	cdc15_290	73	diaux2
22	Alpha119	48	elu000	74	diaux3
23	cdc15_010	49	elu030	75	diaux4
24	cdc15_030	50	elu060	76	diaux5
25	cdc15_050	51	elu090	77	diaux6
26	cdc15_070	52	elu120	78	diaux7

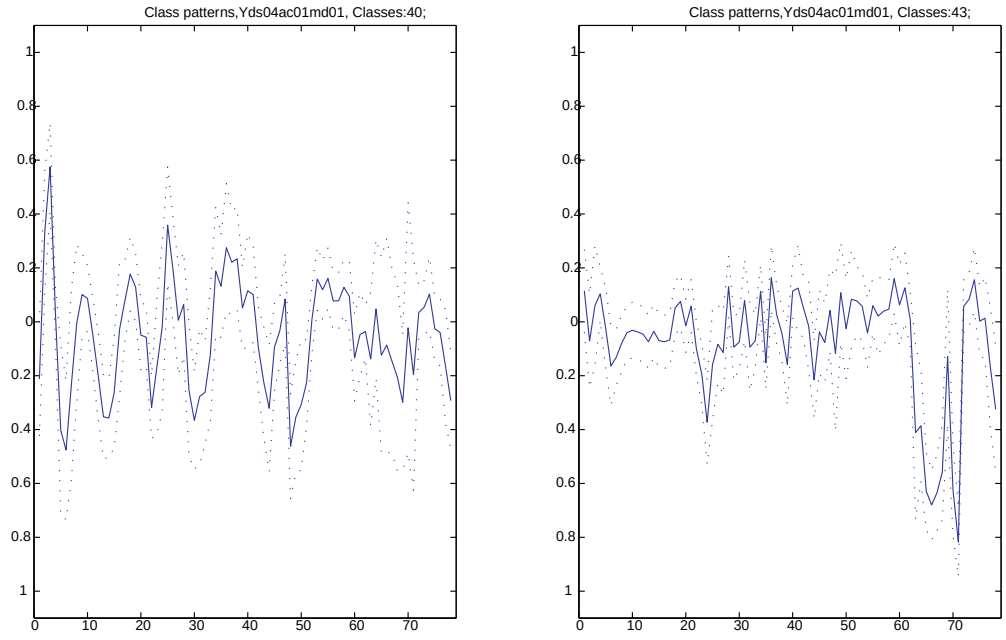
Gene Attribute Vector with Corresponding Experiments

Our classification was on the genes using the experiments/samples as attributes of that gene. The immediate result is a grouping of the genes by similarity of expression patterns. 45 statistically distinguishable patterns were found using the AUTOCLASS classification program from the full data set.

Class #	Class Weight	Class #	Class Weight
1	244	24	128
2	228	25	118
3	222	26	115
4	218	27	115
5	210	28	115
6	189	29	109
7	184	30	107
8	173	31	100
9	166	32	98
10	161	33	98
11	158	34	91
12	156	35	88
13	154	36	84
14	152	37	79
15	147	38	76
16	145	39	57
17	145	40	42
18	144	41	39
19	142	42	38
20	142	43	36
21	139	44	36
22	135	45	33
23	131		

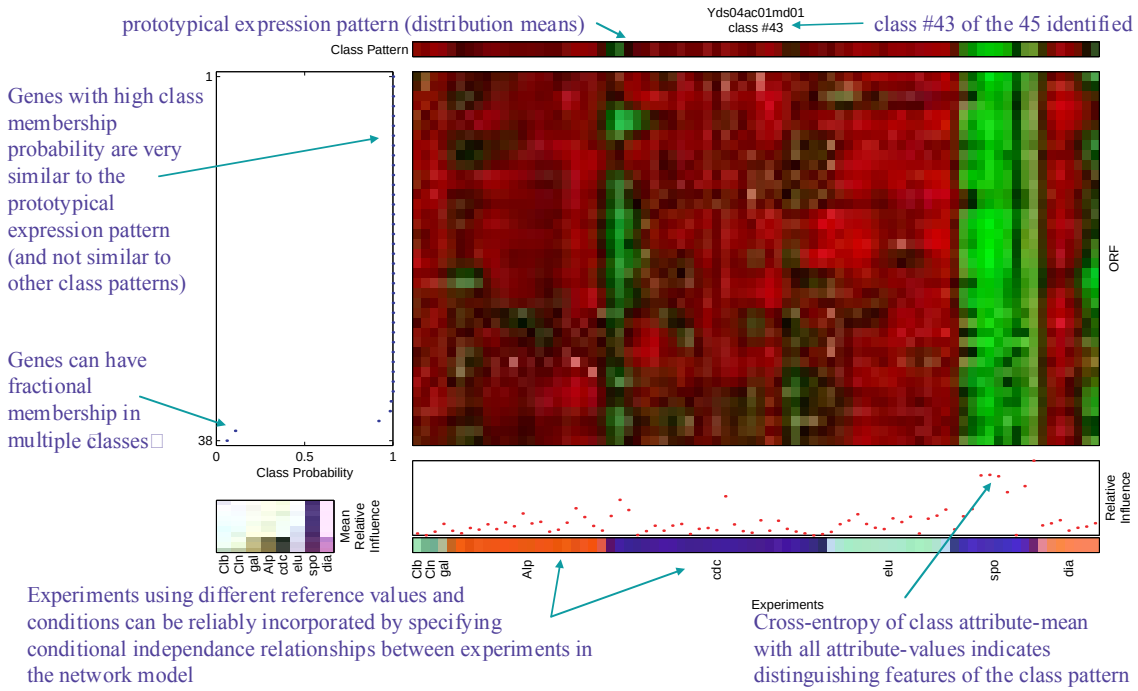
Class Weight for each class

Autoclass identified 45 patterns of expression. The class weight is the sum of the probabilities of membership in that class over all genes. If all genes had a probability of only 1 or 0, then the class weight reduces to the numberof genes in the class.



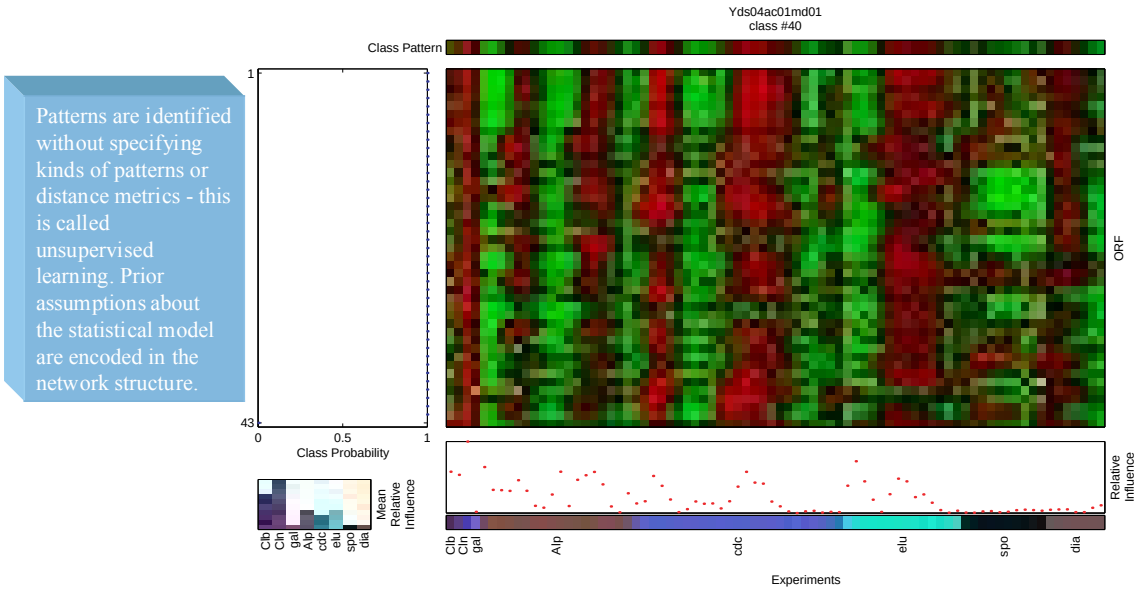
Expression Pattern Distributions Across Experiments for Classes 43 and 45

The horizontal axis is experiment number. The vertical axis is expression level. The solid plot represents the mean at each experiment. The dotted lines represent the width of the distribution of expression levels , 1-sigma from the mean.



New Methods for Gene Expression Analysis and Display

The figure was generated using our tools and includes several novel and informative features. We can generate a display like this for each of the classes identified. This figure shows results for class number 43 of the 45 identified. The large image in the middle represents the expression levels in the manner of *(Eisen 1998)*. Each row represents a gene and each column an experiment/chip. Red indicates increased expression and green indicates decreased expression compared to some reference state. The genes have been reordered based on their probability of belonging to this class-pattern in descending order. The experiments are grouped by study. The colored bars across the bottom of the figure show this grouping of experiments. The alpha-arrest, elutriation, cdc15, sporulation, and diauxic shift studies are all time series. This initial model does not make use of any temporal relationships and the columns can be ordered arbitrarily. Across the top is the prototypical pattern that defines this class. It is the mean value of the PDF in this class for each experiment. The width of the distribution is not represented here, but is used for the joint-density analysis discussed below. Across the bottom is a relative influence term which shows which experiments/attributes distinguish this class from the rest of the data set. This graph allows one to find the experiments which generated the most distinguishing features of this pattern. The y-axis is the cross-entropy between the class PDF and the PDF of a single class model describing the entire data set. Down the left side is a plot of the class membership probability for each gene. It provides a measure of the similarity of each gene to the class expression distribution. The y-scale of this plot shows how many genes are displayed. Only genes with a probability greater than 10% are shown in this figure. Note that the class probability is continuous and allows a gene to belong to more than one expression pattern class or to no class. The small box in the lower left is a bar chart of the mean relative influence of each study on distinguishing this pattern from the rest of the data set. The blue text and green arrows are only included in this figure to label the important features of the display.



Gene Expression Pattern Displayed by Class Membership

specific cell cycle modulations dominate this pattern but was identified without specifying a cyclic structure as input to the search

Several of the identified patterns capture cell cycle modulations with different peak times and phases. The figure shows one such pattern which is referred to as class #40. There are 42 genes which fall strongly into this class (and no other) and hence have a pattern very similar to each other. The relative influence shows the peaks and dips at certain points in the cell cycle are the distinguishing features of this pattern.

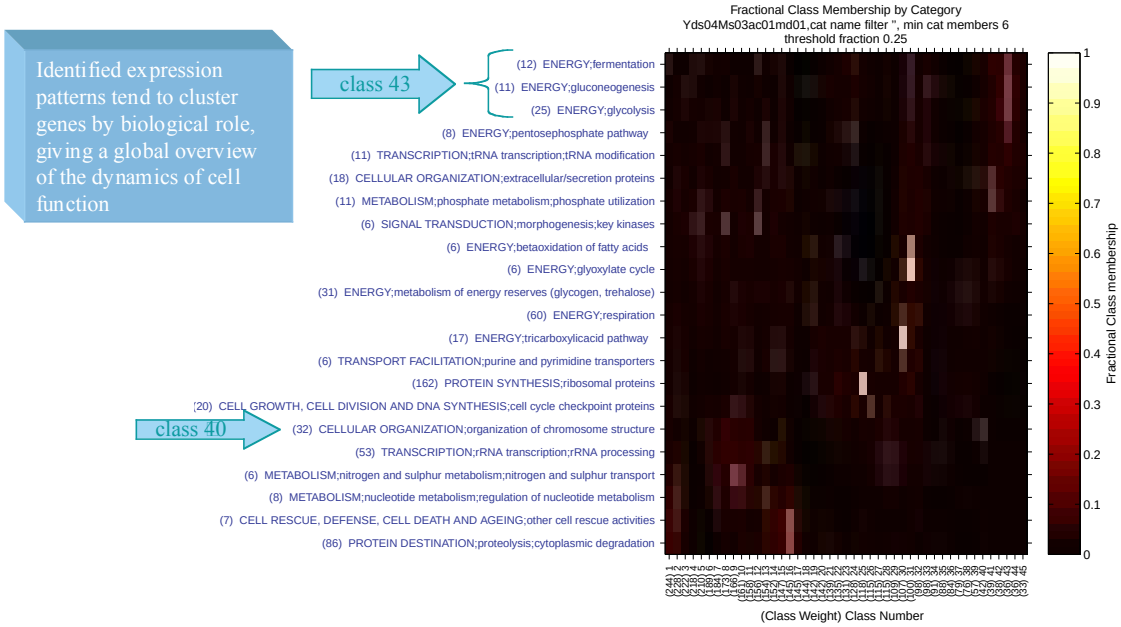
	ORF	Gene	Protein Function (YPD)
chromatin reorganization and structure	YPL127C	HHO1	Histone H1;chromatin/chromosome structure
	YBL003C	HTA2	Histone H2A
	YDR225W	HTA1	Histone H2A
	YBL002W	HTB2	Histone H2B
	YDR224C	HTB1	Histone H2B
	YBR010W	HHT1	Histone H3
	YNL031C	HHT2	Histone H3
	YBR009C	HHF1	Histone H4
	YNL030W	HHF2	Histone H4
	YER095W	RAD51	DNA repair and recombination
cell cyclins	YMR199W	CLN1	G1/S-specific cyclin that interacts with Cdc28p protein kinase to control events at START
	YPL256C	CLN2	G1/S-specific cyclin, interacts with Cdc28p protein kinase to control events at START
	YIL123W	SIM1	Protein involved in the aging process and in regulation of the cell cycle
cell wall generation and maintenance	YDL055C	PSA1	may supply cell-wall precursors for budding, known cell-cycle regulated;cell wall generation and maintenance
	YIL140W	SRO4	Membrane glycoprotein localized at site of bud emergence, required for axial budding pattern
	YJL158C	CIS3	protein localized at surface of growing buds
	YLR300W	EXG1	beta-1,3-glucanase,major isoform involved in cell wall beta-glucan assembly
	YLR342W	FKS1	Component of beta-1,3-glucan synthase
	YMR307W	GAS1	may cross-link glucans and chitin;Glycophospholipid-anchored surface glycoprotein
	YMR215W	GAS3	specific function unknown
	YMR305C	SCW10	hydrolase
	YNL283C	WSC2	Protein required for maintenence of cell wall integrity and for the stress response
carbohydrate metabolism	YOL007C	CSI2	Protein involved in chitin synthesis
	YER001W	MNN1	carbohydrate metabolism;membrane protein;Alpha-1,3-mannosyltransferase
	YFL045C	SEC53	Phosphomannomutase, involved in the synthesis of GDP-mannose and dolichol-phosphate-mannose
	YER070W	RNR1	Nucleotide metabolism;Ribonucleotide reductase (ribonucleoside-diphosphate reductase) large subunit
	YGL225W	GOG5	Golgi GDP-mannose transporter, in nucleotide-sugar transporter (NST) family of membrane transporters
	YKL008C	LAC1	Protein required with Lag1p for ER-to-Golgi transport of GPI-anchored proteins
	YER003C	PMI40	protein modification;Mannose-6-phosphate isomerase
	YLR121C	YPS3	GPI-anchored aspartyl protease
	YML027W	YOX1	Homeodomain protein that binds leu-tRNA gene
unknown	YOR248W		function unknown
	YPL163C	SVS1	specific function unknown;Serine- and threonine-rich protein required for vanadate resistance
	YOR247W	SRL1	function unknown;Protein with similarity to Svs1p
	YKR012C		function unknown
	YGR189C	CRH1	function unknown; resides in chitin rich areas of cell wall
	YKR013W	PRY2	function unknown;Protein expressed under starvation conditions
	YNL300W		function unknown;Protein with weak similarity to Mid2p
	YNR009W		function unknown
	YOL019W		function unknown
	YDR451C		function unknown;Protein with similarity to bacterial leucyl aminopeptidase

Genes with >90% probability of belonging to class 40

Genes can be sorted based on class membership and inspected for similarities in function. We provide more quantitative means to do this analysis is shown below.

Integrating Knowledge Bases

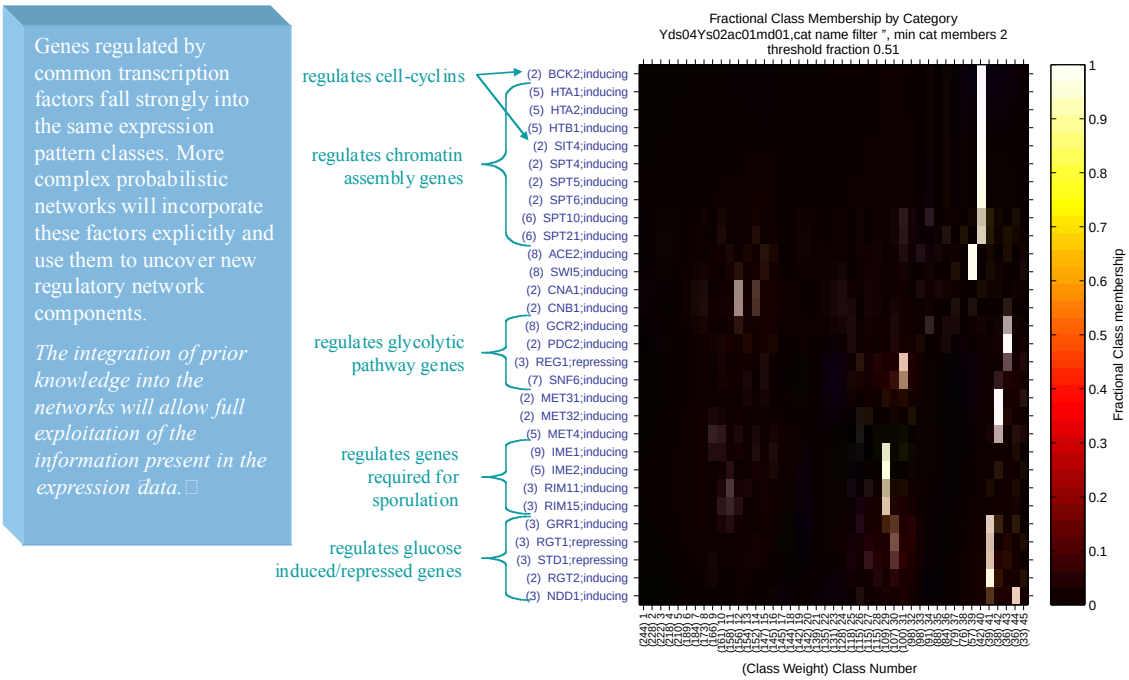
The classes can be used in combination with other information to extract information on cellular dynamics. Integrating the expression data and classes with other knowledge bases is facilitated by the use of the gene-attribute vector and the classification results, which groups genes according to likelihood of belonging to one of many attribute-value patterns identified. One method of integrating the expression classes with categorical databases, and thus assign some meaning to the classes found, is to look for enrichment of a functional category in a single class. An important feature to our approach is that a gene can be labeled as belonging to more than one functional category and class.



Fraction of Genes in Each Class by MIPS Category

In the figures, orange-white represents a strong enrichment of genes of a particular category in a class. The categories are displayed along the vertical axis and the classes along the horizontal axis. The categories are reordered to place those with similar class memberships together. The MIPS-database category is a yeast gene annotation describing the biological function or role of

the encoded protein*(Mewes 1999)*. The transcription factors were extracted from the YPD database*(Hodges 1999)*. Due to the large number of classes and categories, we use filters to display only the most significant categories or members. In the figure above, only categories with at least 6 genes and which have at least one class fraction of >= 25% are displayed.



Gene Transcription Factors and Class Membership

Inference

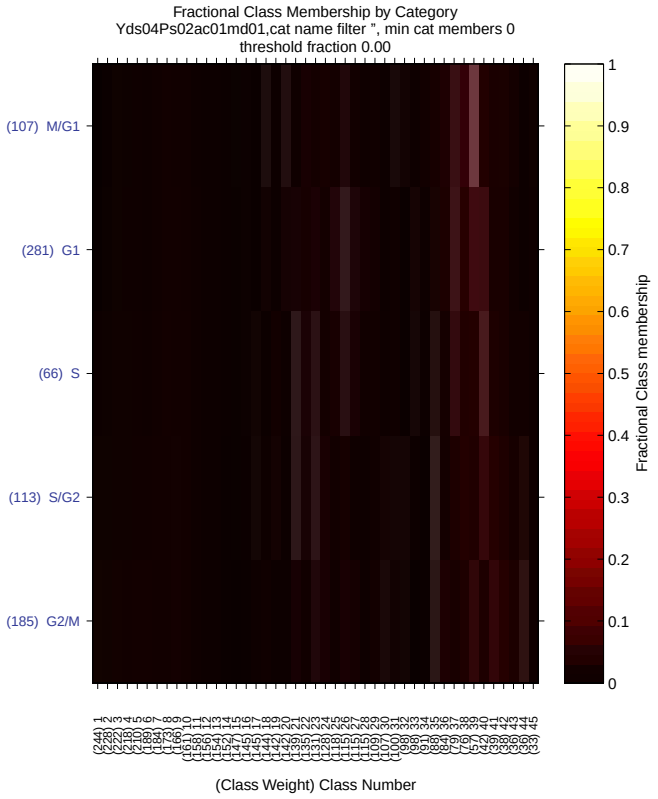
Using class membership probabilities and the cross-indexing information, we can associate the function of known genes with those whose functions are not assigned. For example, class 25 of the yeast gene model contains a very high fraction of ribosomal proteins. If we examine the genes with membership probabilities which are >10% we find 120 genes. Of those genes, only 4 are not readily identifiable as ribosomal proteins or translation factors. However, 11 genes are of unknown function or only annotated by interspecies sequence homology.

ORF	Gene	Class 25 membership prob.	MIPS Brief ID (updated 11/30/99)
YDR417C	NONE	1.00	questionable ORF
YGL102C	NONE	1.00	questionable ORF
YJL188C	NONE	1.00	questionable ORF
YKL056C	NONE	1.00	strong similarity to human IgE-dependent histamine-releasing factor
YLL044W	NONE	1.00	
YLR061W	NONE	1.00	questionable ORF
YLR076C	NONE	1.00	questionable ORF
YNL119W	NONE	1.00	weak similarity to M.jannaschii hypothetical protein MJ1257
YPL142C	NONE	1.00	questionable ORF
YEL026W	NONE	1.00	strong similarity to high mobility group-like protein Nhp2p
YHR193C	EGD2	1.00	similarity to human alpha-NAC
YLR339C	NONE	1.00	questionable ORF
YLR150W	STM1	1.00	specific affinity for guanine-rich quadruplex nucleic acids
YPL037C	EGD1	1.00	GAL4 DNA-binding enhancer protein
YLR293C	GSP1	1.00	GTP-binding protein of the ras superfamily
YAL038W	CDC19	0.94	pyruvate kinase

Selected genes from class 25

Gray entries are genes with unknown function. Yellow entries have assigned functions, but are not ribosomal. There are 105 ribosomal gene in this class (not shown).

Finally, cross-indexing the cell-cycle indexed genes from *(Spellman 1998)* allows a comparison of our classes to those determined by Fourier analysis. There are several classes which correspond very well to cell-cycle peak assignments. The splitting into several classes is not surprising considering the number of experiments included in classification which were not cell-cycle experiments. For example, the G1 genes are mostly split between classes 26 and 37. Inspection of their class patterns and attribute influences show that the genes are modulated together in cell-cycle experiments and diverge during sporulation.



Applications to tumorous and non-tumorous Colon Tissue Gene Expression

Expression data for normal and cancerous colon tissues *(Alon 1999)* were classified by patient and tumor type using the expression values of the genes as an attribute vector. There are 62 samples including 40 cancerous and 22 non-cancerous ones. There are 1988 genes measured per sample. The colon tissue data were normalized to the mean value of each chip as described in the original publication*(Alon 1999)*. In contrast to other methods, no pre-selection of the data was necessary, nor did we shift or rescale the expression patterns across experiments.

We have identified a set of significantly different expression patterns which group the tissue samples. Four classes were identified based only upon gene expression data. The table shows the make-up of each class with each tissue. The class members are collected in the colored boxes. The gray colored samples in the list are non-tumorous tissue samples. Classes 1 and 4 consist almost entirely of tumorous samples. These classes are identified based only on gene-expression and did not include the tumorous/non-tumorous label. Therefore, the classification identifies two gene-expression patterns indicative of tumorous cells.

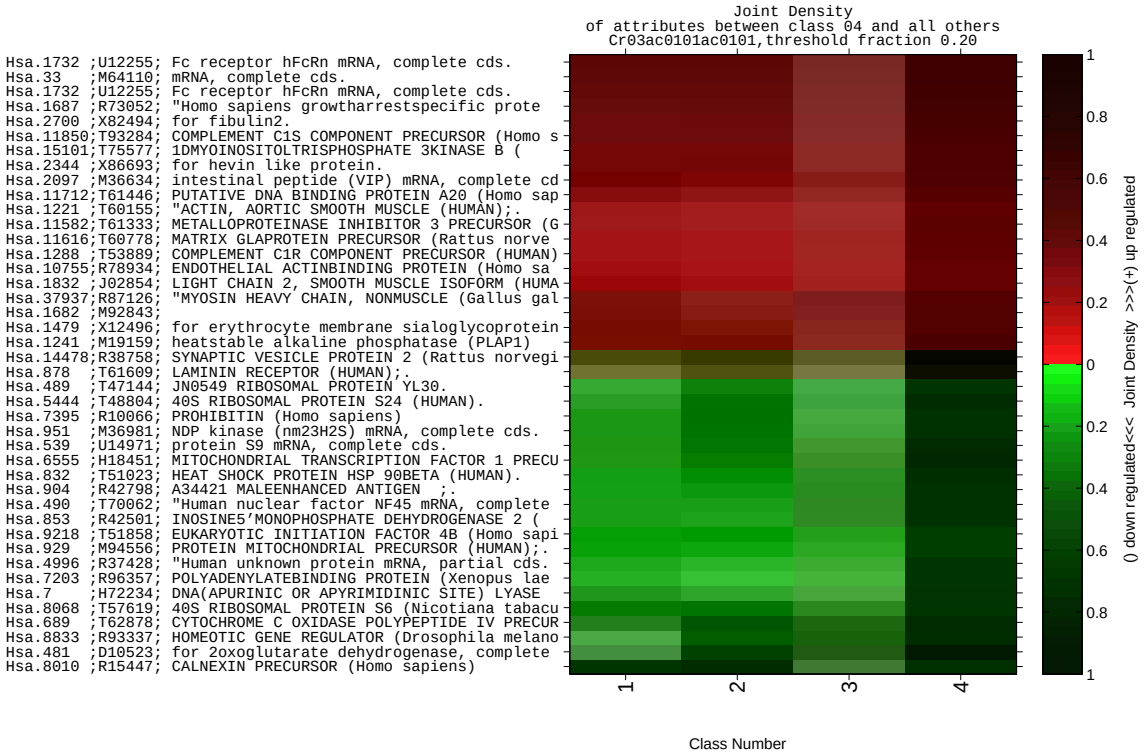
ORF	class 1	class 2	class 3	class 4
tumor;patient_01	1	0	0	0
tumor;patient_02	1	0	0	0
normal;patient_02	1	0	0	0
tumor;patient_04	1	0	0	0
tumor;patient_06	1	0	0	0
normal;patient_06	1	0	0	0
tumor;patient_11	1	0	0	0
tumor;patient_14	1	0	0	0
tumor;patient_15	1	0	0	0
tumor;patient_16	1	0	0	0
tumor;patient_17	1	0	0	0
tumor;patient_18	1	0	0	0
tumor;patient_19	1	0	0	0
tumor;patient_20	1	0	0	0
tumor;patient_22	1	0	0	0
tumor;patient_23	1	0	0	0
tumor;patient_25	1	0	0	0
tumor;patient_27	1	0	0	0
tumor;patient_28	1	0	0	0
tumor;patient_37	1	0	0	0
tumor;patient_38	1	0	0	0
tumor;patient_03	0	1	0	0
normal;patient_03	0	1	0	0
normal;patient_04	0	1	0	0
tumor;patient_05	0	1	0	0
normal;patient_05	0	1	0	0
tumor;patient_07	0	1	0	0
normal;patient_07	0	1	0	0
tumor;patient_08	0	1	0	0
normal;patient_08	0	1	0	0
tumor;patient_09	0	1	0	0
normal;patient_09	0	1	0	0
tumor;patient_10	0	1	0	0
normal;patient_10	0	1	0	0
tumor;patient_12	0	1	0	0
normal;patient_12	0	1	0	0
normal;patient_27	0	1	0	0
tumor;patient_35	0	1	0	0
normal;patient_01	0	0	1	0
normal;patient_11	0	0	1	0
normal;patient_28	0	0	1	0
normal;patient_29	0	0	1	0
tumor;patient_30	0	0	1	0
normal;patient_32	0	0	1	0
tumor;patient_33	0	0	1	0
normal;patient_33	0	0	1	0
normal;patient_34	0	0	1	0
normal;patient_35	0	0	1	0
normal;patient_36	0	0	1	0
tumor;patient_36	0	0	1	0
normal;patient_39	0	0	1	0
tumor;patient_40	0	0	1	0
normal;patient_40	0	0	1	0
tumor;patient_13	0	0	0	1
tumor;patient_21	0	0	0	1
tumor;patient_24	0	0	0	1
tumor;patient_26	0	0	0	1
tumor;patient_29	0	0	0	1
tumor;patient_31	0	0	0	1
tumor;patient_32	0	0	0	1
tumor;patient_34	0	0	0	1
tumor;patient_39	0	0	0	1

Class Membership for Colon Tissue Gene Expression Classes

The colored blocks indicate classes of samples. Gray boxes indicate non-tumorous colon tissue. Other samples are tumorous tissue. Note that classes 1 and 4 (yellow and orange) are both almost entirely tumorous samples.

Which genes are responsible for the distinction between the two tumor classes? We can use statistical tests on the classification results to identify which genes (attributes) were most significant in distinguishing the classes (of samples) from each other, thereby gaining insight into

the important players in the tumorous samples. We calculate the joint probability between classes for each attribute. A joint probability of zero indicates gene expression levels which are completely separated and distinct, while a joint probability of one indicates identical distributions of expression levels. See the appendix for a more detailed description of the joint probability.

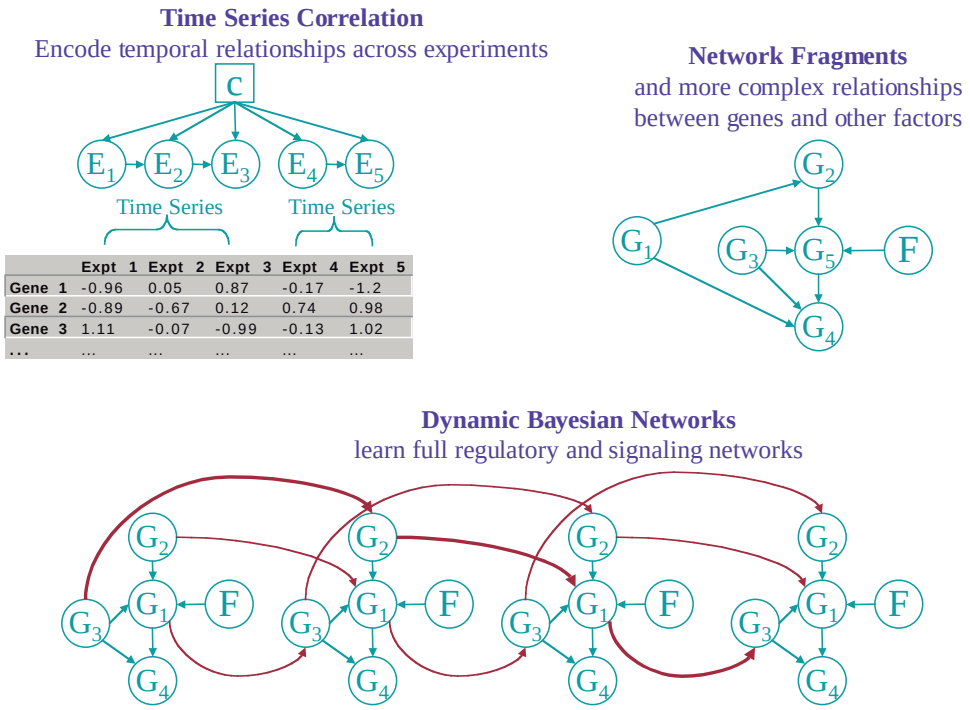


The figure displays the attributes that are most significant in distinguishing class 4 from the other classes. The more intense colors represent highly distinguishing genes, i.e. a smaller joint density between the distributions. The sign indicates the direction of change of the mean expression level: positive (red) indicates an up-regulation relative to the class 4 distribution, negative (green) indicates down-regulation relative to class 4.

Future Work

Model Extension

We are pursuing further extensions of these methods to more complex statistical descriptions. The statistical models described so far have only used real-valued variables described by normal distributions to describe the observations. The software toolboxes already accommodate mixtures with discrete variables, which can be useful for including information such as blood type or transposon site. The models we have described so far are simple in terms of network structure and the statistical correlations encoded. We are extending the methods to dynamic Bayesian networks to further develop hypothesis testing and reverse engineering applications.



Bold arrows highlight a regulatory loop over time

Extension to more complex networks

Model Queries

We are developing additional methods to extract useful information from the data models, including finding joint probabilities of various kinds. In addition, we are exploring supervised classification methods labelled nodes, e.g. training with a pre-determined number of classes with observed labels such as 'ALL' and 'AML'.

Implementation Issues

Finding the optimal solution to the network structure and parameter values is a computationally demanding task. Also of concern is computing the results of queries to the network - Bayesian estimation from the model requires the calculation of large, multi-dimensional integrals. Ideally, other constraints can be put on the network parameters and structure to allow further probabilistic inference. The scalability of the methods with more data samples, more variables, and network complexity is essential. We have implemented multidimensional adaptive Monte Carlo methods based on the VEGAS algorithm*(Press 1995)*.

References

Alon, U., Barkai, N., Notterman, D.A., Gish, K., et al. (1999). "*Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*." Proceedings of the National Academy of Sciences of the United States of America **96(12)**: 6745-6750.

Cheeseman, P., Kellay, J.K., Self, M., Stutz, J., et al. (1990). "*Autoclass: A Bayesian classification system*". Readings in Machine Learning. San Mateo, CA, Morgan Kaufmann Publishers**:** 296-306.

Chen, T., He, H.L. and Church, G.M. (1999). "*Modeling gene expression with differential equations*." Pacific Symposium on Biocomputing **120(7)**: 29-40.

Chu, S., DeRisi, J., Eisen, M., Mulholland, J., et al. (1998). "*The transcriptional program of sporulation in budding yeast*." Science **282(5389)**: 699-705.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). "*Maximum likelihood from incomplete data via the EM algorithm*." Journal of the Royal Statistical Society, Series B **39(1)**: 1-38.

DeRisi, J.L., Iyer, V.R. and Brown, P.O. (1997). "*Exploring the metabolic and genetic control of gene expression on a genomic scale*." Science **278(5338)**: 680-6.

Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. (1998). "*Cluster analysis and display of genome-wide expression patterns*." Proceedings of the National Academy of Sciences of the United States of America **95(25)**: 14863-14868.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., et al. (1999). "*Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring*." Science **286(5439)**: 531-537.

Hanson, R., Stutz, J. and Cheeseman, P. (1990). *Bayesian Classification Theory*, NASA Ames Research Center.

Heckerman, D. (1997). "*Bayesian Networks for Data Mining*." Data Mining and Knowledge Discovery **1**: 79-119.

Hodges, P.E., McKee, A.H.Z., Davis, B.P., Payne, W.E., et al. (1999). "*Yeast Proteome Database (YPD): a model for the organization and presentation of genome-wide functional data.*" Nucleic Acids Research **27**: 69-73.

Jensen, F. (1998). An Introduction to Bayesian Networks. Santa Clara, Springer.

Liang, S., Fuhrman, S. and Somogyi, R. (1998). "*Reveal, a general reverse engineering algorithm for inference of genetic network architectures*." Pacific Symposium on Biocomputing **95(1)**: 18-29.

Mewes, H., Heumann, K., Kaps, A., Mayer, K., et al. (1999). "*MIPS: a database for protein sequences and cpmplete genomes*." Nucleic Acids Research **27**: 44-48.

Perou, C.M., Jeffrey, S.S., van de Rijn, M., Rees, C.A., et al. (1999). "*Distinctive gene expression patterns in human mammary epithelial cells and breast cancers*." Proceedings of the National Academy of Sciences of the United States of America **96(16)**: 9212-7.

Potts, J.T. (1996). *Seeking Parallelism in Discovery Programs*, The University of Texas at Arlington.

Press, W., Teukolsky, S., Vetterling, W. and Rannery, B. (1995). Numerical Recipes in C. New York, Cambridge University Press.

Spellman, P.T., Sherlock, G., Futcher, B., Brown, P.O., et al. (1998). "*Identification of cell cycle regulated genes in yeast by DNA microarray hybridization*." Molecular Biology of the Cell **9(SUPPL.)**: 371A.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., et al. (1998). "*Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization*." Molecular Biology of the Cell **9(12)**: 3273-97.

Supplement, S. (Jan 1999). Nature Genetics **21(1 Supplement)**.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., et al. (1999). "*Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation*." Proceedings of the National Academy of Sciences of the United States of America **96(6)**: 2907-12.

Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., et al. (1999). "*Systematic determination of genetic network architecture [see comments]*." Nature Genetics **22(3)**: 281-5.

Titterington, D.M., Smith, A.F.M. and Makov, U.E. (1985). Statistical Analysis of Finite Mixture Distributions. New York, John Wiley & Sons.

Weaver, D.C., Workman, C.T. and Stormo, G.D. (1999). "*Modeling regulatory networks with weight matrices*." Pacific Symposium on Biocomputing **94(23)**: 112-23.

Appendices

Finite Mixture Model - Principal Equations

The following discussion is based largely on the references *(Cheeseman 1990; Hanson 1990; Potts 1996).*

S is a finite space of all possible models

h is a particular model, embedded in S, characterized by:

a set of classes C with J members, $c_1 \ldots c_J$

an inter-class probability distribution function τ and it's parameters υ, specifying $p(c_j \mid \upsilon, \tau, h, S)$

Each class specifies a set of probability distribution functionals T and their parameters V for each of K attributes, $T_j : t_{j1\ldots} t_{jK}$ and $V_j : v_{j1\ldots} v_{jK}$

The observed data set is E with I cases of observations $E_1 \ldots E_I$ of the K attributes, $e_{1,1} \ldots e_{I,K}$

Gene Attribute Vector

The gene attribute vector used for learning the model is, at minimum, a set of real-numbered expression levels. These attributes have an uncertainty associated with the experimental measurements. The uncertainty in the data is accounted for in the learning procedure by not allowing the width of the class-attribute pdf to be less than this uncertainty. The attribute-vector can also include other types of information, such as category membership, alternative pdfs, etc. Augmenting the expression patterns with such information is potentially very useful. We call a particular observation of the attribute vector an attribute instance. The results of the classification are a set of identified attribute-value patterns, called classes, which have a high likelihood of describing naturally occuring groupings in the attribute vectors observed.

Probability Distribution Functions

Each class describes a probability distribution for each attribute. For example, if t_{jk} specifies a normal distribution over real number attribute k for class j, then the probability of observing the ith case datum for that attribute e_{ik} would be defined by the normal functional form

$$P(e_{ik} \mid c_j, v_{jk}, t_{jk}) = \frac{1}{\sqrt{2\pi\sigma_{jk}}} e^{-\frac{1}{2}\left[\frac{e_{ik}-\mu_{jk}}{\sigma_{jk}}\right]^2}$$

with specified parameters

$$v_{jk} : \sigma_{jk}, \mu_{jk}$$

Similarly, pdfs for enumerated data types and other distributions can be specified. Currently we are only working with real-valued data and normal distributions.

Likelihood of Observing the Data

The probability of observing a single case, given the class, the pdf form and it's parameters, is a product over all attributes

$$P(E_i \mid c_j, V_j, T_j) = \prod_k P(e_{ik} \mid c_j, v_{jk}, t_{jk})$$

The joint probability of the data with the class, given the model and all of the model parameters and specifications, is

$$\begin{aligned} P(E_i, c_j \mid V_j, T_j, \nu, \tau, h, S) &= P(c_j \mid \nu, \tau, h, S) P(E_i \mid c_j, V_j, T_j) \\ &= P(c_j \mid \nu, \tau, h, S) \prod_k P(e_{ik} \mid c_j, v_{jk}, t_{jk}) \end{aligned}$$

The total probability of observing a single case, then, is the sum of the joint over all classes

$$\begin{aligned} P(E_i \mid V, T, C, \nu, \tau, h, S) &= \sum_j P(c_j \mid \nu, \tau, h, S) P(E_i \mid V_j, T_j) \\ &= \sum_j \left[P(c_j \mid \nu, \tau, h, S) \prod_k P(e_{ik} \mid v_{jk}, t_{jk}) \right] \end{aligned}$$

and the total probability of observing the entire data set, given the model, is the product of the probabilities of each observation

$$P(E \mid V, T, C, \upsilon, \tau, h, S) = \prod_i \sum_j \left[P(c_j \mid \nu, \tau, h, S) \prod_k P(e_{ik} \mid v_{jk}, t_{jk}) \right]$$

Learning the Model

Learning the model involves a two-level search. The highest level is to find the MAP model form conditioned on the data

$$P(T, C, \tau, h \mid E, S) = \frac{P(T, C, \tau, h \mid S) P(E \mid T, C, \tau, h, S)}{P(E \mid S)}$$

Assuming $P(E \mid S)$ is a constant. We also introduce a uniform model prior on the probability of the model given the model space: all models are equally likely *a priori*. This allows us to simplify the above equation to

$$P(T,C,\tau,h \mid E,S) \propto P(E \mid T,C,\tau,h,S) = \iint dVd\nu P(E,V,\upsilon \mid T,C,\tau,h,S)$$

The lower level search is to find the MAP parameter values, conditioned on the data, given the model form

$$P(V,\upsilon \mid E,T,C,\tau,h,S) = \frac{P(E,V,\nu \mid T,C,\tau,h,S)}{P(E \mid T,C,\tau,h,S)} = \frac{P(E,V,\nu \mid T,C,\tau,h,S)}{\iint dVd\nu P(E,V,\nu \mid T,C,\tau,h,S)}$$

The problem is to find $P(E,V,\nu \mid T,C,\tau,h,S)$ and evaluate its integral over all possible parameter values. We can explicitly write out the form of this equation

$$\begin{aligned} P(E,V,\nu \mid T,C,\tau,h,S) &= P(V,\upsilon \mid T,C,\tau,h,S)P(E \mid V,T,C,\upsilon,\tau,h,S) \\ &= P(V,\upsilon \mid T,C,\tau,h,S)\prod_i \sum_j P(E_i,C_j \mid V_j,T_j,h,S) \\ &= P(V,\upsilon \mid T,C,\tau,h,S)\prod_i \left[\sum_j P(c_j \mid \nu,\tau,h,S)\prod_k P(e_{ik} \mid v_{jk},t_{jk})\right] \end{aligned}$$

We again assume a minimum information form for the prior expectations on the parameters

$$P(V,\upsilon \mid T,C,\tau,h,S) = P(\upsilon \mid \tau,h,S)\prod_{jk} P(v_{jk} \mid t_{jk},h,S)$$

There is an implicit penalty for adding more classes into the model which is represented by $P(c_j \mid \nu,\tau,h,S)$. Because the sum of all class probabilities must be unity, increasing the number of classes lowers the prior probability of each class. Unless the additional classes lead to a higher probability of the observations, the joint will be smaller.

We maximize the joint probability $P(E,V,\nu \mid T,C,\tau,h,S)$ using a variation of the EM algorithm of Dempster and Titterington*(Dempster 1977; Titterington 1985)* with one additional assumption: that each case in the training set belongs to some class. This allows us to use a normalized class membership probability $P(E_i,C_j \mid V_j,T_j,h,S)$ to update the parameter estimates.

The algorithm is

1. Start with guessed parameters V
2. evaluate $P(E_i,C_j \mid V_j,T_j,h,S)$ explicitly
3. re-estimate parameters V using a sum weighted over class membership, e.g.

$$\mu'_{jk} = \frac{\sum_i P(e_{ik}, c_j)\mu_{jk}}{\sum_i P(e_{ik}, c_j)}$$

4. plug the resulting parameter estimates to evaluate the joint probability in step 2 and repeat

The algorithm will find a local maximum in the joint probability and thus in the MAP parameter values. Because there are many local minima, the learning algorithm must guess many initial parameter values and optimize. The overall model fitness for each optimum set of parameters may be evaluated by the integral of $P(E,V,\nu \mid T,C,\tau,h,S)$. The evaluation of this integral is difficult because of the high dimensionality of the parameter space and must be approximated. The integral is approximated by making use of the fact that the best optima of the model form and parameters result in an integral over the joint probability which is dominated by the integral over the region of the optimum parameters

$$\iint dVd\nu P(E,V,\nu \mid T,C,\tau,h,S) \sim \iint_R P(E,V,\nu \mid T,C,\tau,h,S)$$

Where the region R is a region surrounding the locally maximum parameter values. This integral is reported as the model marginal and can be used to find the best models.

The Attribute Influence

The attribute influence A_{jk} is a useful parameter for distinguishing which attributes best distinguish the class from the data set as a whole. This is done by taking the cross-entropy of the model in question with a single class model h_0 which describes the entire data set.

$$A_{jk} = \sum_i P(e_{ik} \mid v_{jk}, t_{jk}, c_j, h, S) \log \frac{P(e_{ik} \mid v_{jk}, t_{jk}, c_j, h, S)}{P(e_{ik} \mid v_{0k}, t_{0k}, c_0, h_0, S)}$$

T h e M a r g

T h e m a r g
w h h i m c o s t d
t w o p -p r r o o b b
p r o d u *k*. c t W e
u s i n g B a

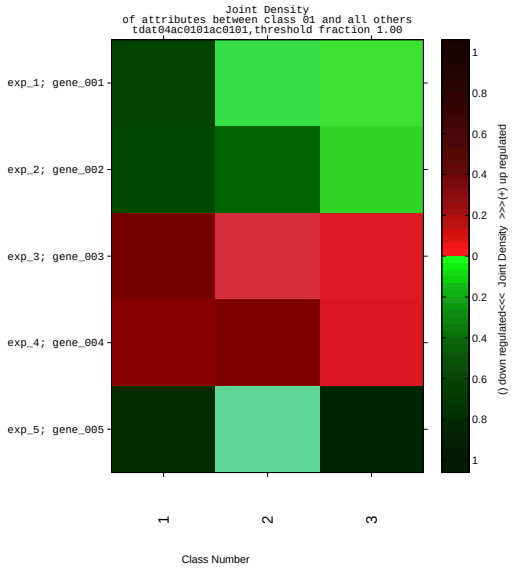
$$
\begin{aligned}
jpcf\left(c_{j1}, c_{j2} \mid k\right) &\equiv P\left(c_{j1}, c_{j2} \mid k, h, S\right) = \int P(c_{j1}, c_{j2}, e_k \mid h, S) de_k \\
&= \int P\left(c_{j1}, c_{j2} \mid e_k, h, S\right) P(e_k \mid h, S)\, de_k \\
&= \int P\left(c_{j1} \mid e_k, c_{j2}, h, S\right) P\left(c_{j2} \mid e_k, h, S\right) P(e_k \mid h, S) de_k \\
&= \int P\left(c_{j1} \mid e_k, h, S\right) P\left(c_{j2} \mid e_k, h, S\right) P(e_k \mid h, S) de_k \\
&= \int \frac{P\left(e_k \mid c_{j1}, h, S\right) P\left(c_{j1} \mid h, S\right)}{P\left(e_k \mid h, S\right)} \cdot \frac{P\left(e_k \mid c_{j2}, h, S\right) P\left(c_{j2} \mid h, S\right)}{P\left(e_k \mid h, S\right)} P(e_k \mid h, S)\, de_k \\
&= \int \frac{P\left(e_k \mid c_{j1}, h, S\right) P\left(e_k \mid c_{j2}, h, S\right)}{J \sum_{c=1}^{J} P\left(e_k \mid c_{jc}, h, S\right)} de_k
\end{aligned}
$$

where e_k is the value for attribute k, only.

The *jpcf* is always between zero and one. We add further information to the display of the *jpcf* by giving it a sign: a positive sign means that the mean of class j2 is higher than the mean of class j1; a negative sign indicates the reverse.

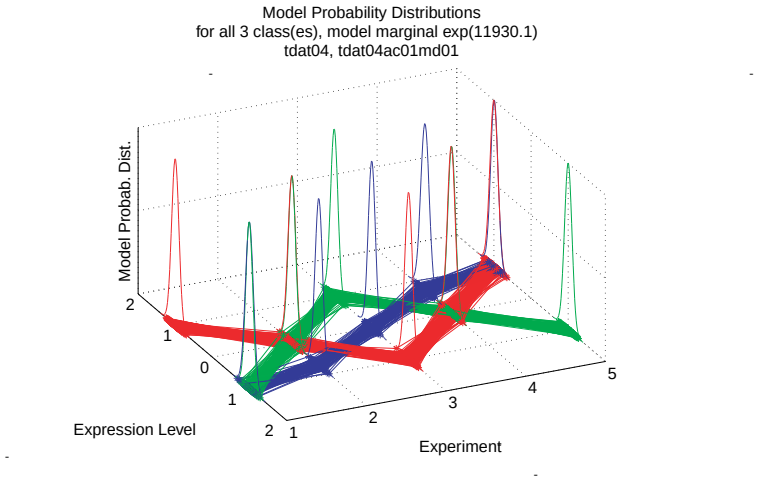
A crude but fast approximation to the integral above can be implemented with implicit importance sampling using the observed data as the samples. In this case, the marginal joint is approximately,

$$
jpcf \approx \frac{1}{N} \sum_{i=1}^{N} \frac{P\left(e_{ik} \mid k, c_{j1}, h, S\right) P\left(e_{ik} \mid k, c_{j2}, h, S\right)}{\left(\sum_{c} P\left(e_{ik} \mid k, c_{jc}, h, S\right)\right)^2}
$$



Jpcf of class 1 vs all other classes by attribute for test data

Brighter colors represent most significant differences. Red indicates that the mean expression level is higher in class n than class 1, green the opposite. Black indicates identical attribute distributions.



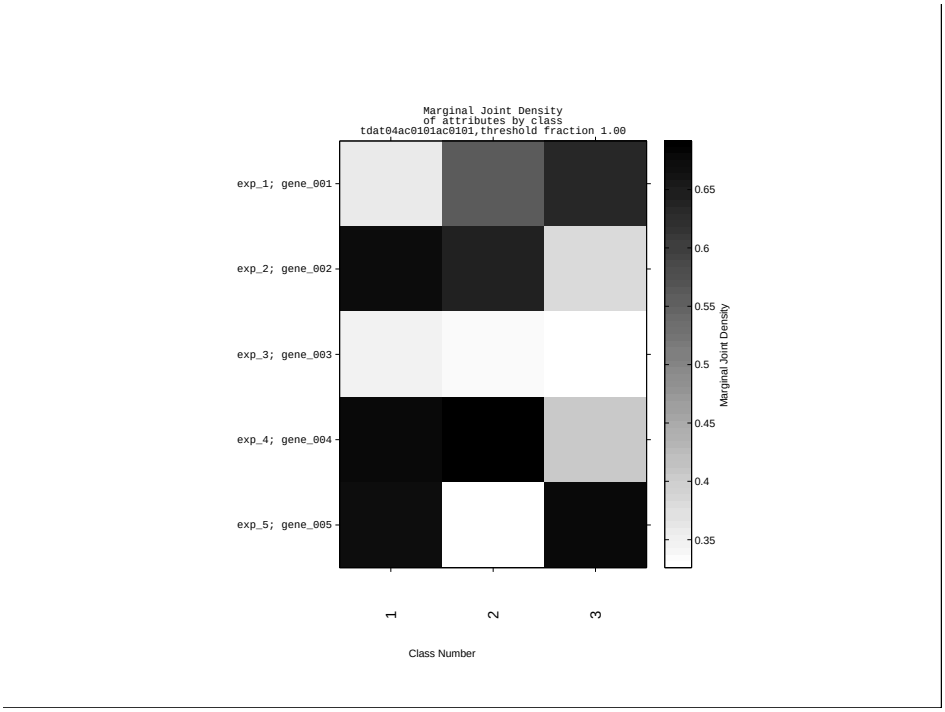
Test data and model distributions

Class number 1 is red. Classes two and three are green and blue, respectively.

We can use the marginal joint probability recursively to find which attributes are most important in distinguishing one class from *all* of the others by calculating the marginal probability for class j1. This summed joint density measure is can be expressed as,

$$
\begin{aligned}
jpcf(c_{j1} \mid k) &\equiv P\left(c_{j1} \mid k, h, S\right) \\
&= \sum_{j2} P\left(c_{j1}, c_{j2} \mid k, h, S\right) \\
&= \frac{1}{J} \sum_{j2}^{J} jpcf\left(c_{j1}, c_{j2} \mid k\right)
\end{aligned}
$$

Those attributes which have a low *jpcf* are the most distinguishing attributes. and the attribute/class combinations with the smallest values of *jpcf* are the most important.



Similarly, we can compare the patterns of overlap between classes using the *jpc* by using all of the attributes,

$$
\begin{aligned}
jpcf\left(c_{j1}, c_{j2}\right) &\equiv P\left(c_{j1}, c_{j2} \mid, h, S\right) = \\
&= \int \frac{P\left(e \mid c_{j1}, h, S\right) P\left(c_{j1} \mid h, S\right)}{P\left(e \mid h, S\right)} \cdot \frac{P\left(e \mid c_{j2}, h, S\right) P\left(c_{j2} \mid h, S\right)}{P\left(e \mid h, S\right)} P(e \mid h, S) de \\
&= \int \frac{P\left(e \mid c_{j1}, h, S\right) P\left(e \mid c_{j2}, h, S\right)}{J \sum_{c=1}^{J} P\left(e \mid c_{jc}, h, S\right)} de
\end{aligned}
$$

where

$$P(e \mid c_{jc}, h, S) = \prod_k P(e_k \mid c_{jc}, h, S)$$

Note that the integral in the jpc is multidimensional, spanning the k-dimensional space of attribute values. This integral is estimated by Monte Carlo methods or by implicit importance sampling.

Cross-indexing with categories

One method of integrating the expression classes with categorical databases, and thus assign some meaning to the classes found, is to look for enrichment of a functional category in a single class. We want to find the fraction of genes falling into each class for each category. Combinations of functional categories co-occurring in a class pattern may also be useful. We define the enrichment $R(M,C)$ to be the sum of the class probabilities over all of the genes that are members of the category, normalized so that the sum across all classes for a single category is one. This can be written as a matrix multiplication,

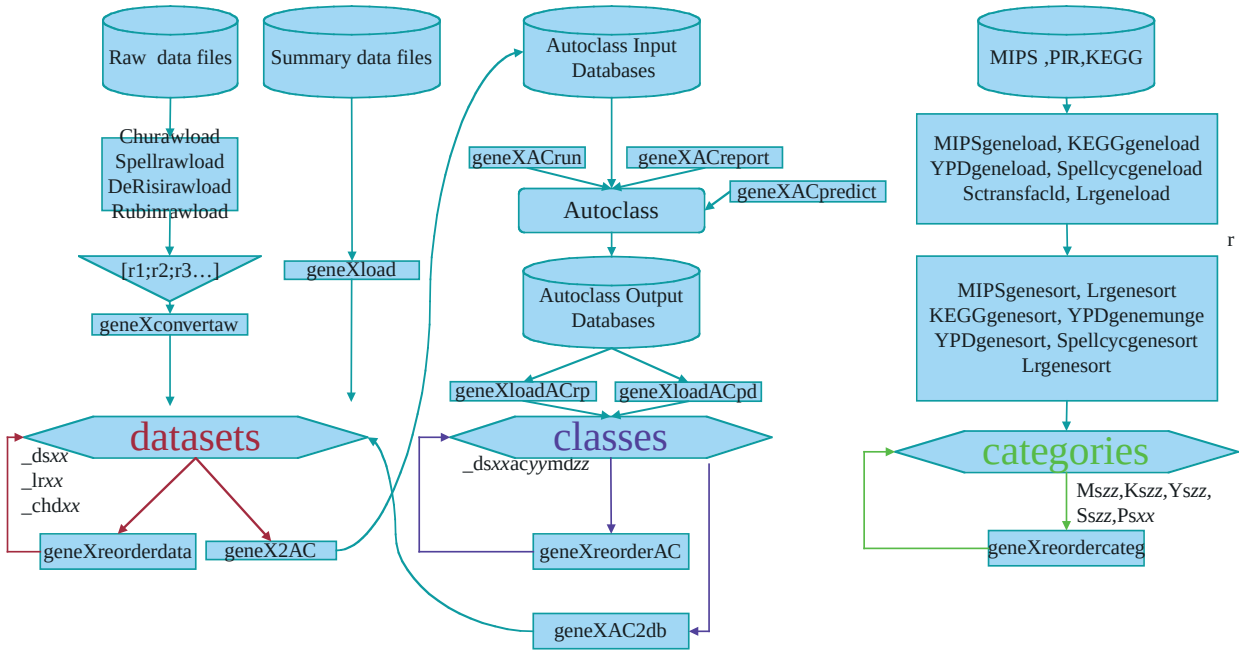
$$R(M,C) = P(E \mid C, V, T\nu, \tau, h, S) \times P(E,M)$$

where $P(E,M)$ is a category membership matrix with each row corresponding to a functional category, each column corresponding to a single gene, and each entry being 1 or 0 for member or non-member.

$$Enrichment = \begin{pmatrix} & class1 & class2 & \cdots \\ categ1 & \cdot & \cdot & \cdot \\ categ2 & \cdot & \cdot & \cdot \\ \cdots & \cdot & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} & gene1 & gene2 & \cdots \\ categ1 & 1 & 0 & \cdot \\ categ2 & 0 & 0 & \cdot \\ \cdots & \cdot & \cdot & \cdot \end{pmatrix} \times \begin{pmatrix} & class1 & class2 & \cdots \\ gene1 & 0.95 & 0.05 & \cdot \\ gene2 & 0 & 0.99 & \cdot \\ \cdots & \cdot & \cdot & \cdot \end{pmatrix}$$

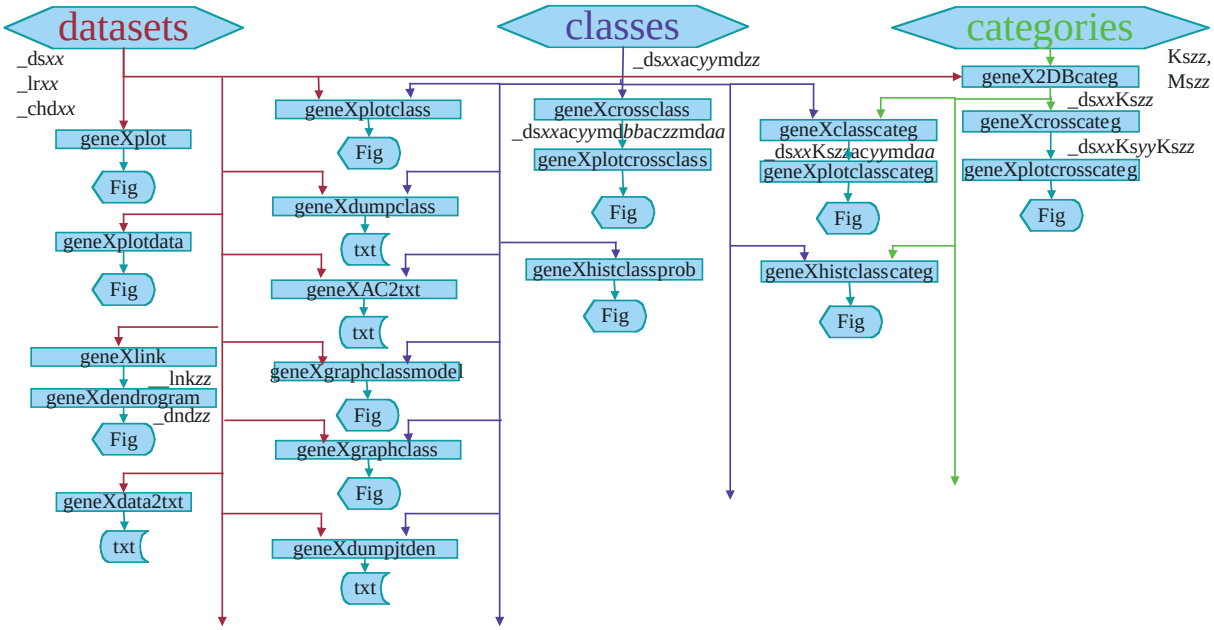
Dataflow and Software

Matlab routines have been written which perform the operations and generate the visualizations described above. There are also routines that create input files for AUTOCLASS and read the output files for further analysis. There are 3 major types of data structures defined: datasets, classes, and categories. Datasets are loaded using specific formats. Classes are results of AUTOCLASS search or predict operations. Categories are databases of labels for genes or samples. There are a variety of routines for combining and sorting these structures and for generating displays of the results.



Software Components and Dataflow into the Internal Storage Objects

Boxes represent collections of matlab routines. Labels in boxes are names of principle functions. Hexagons represent internal storage objects. Round-cornered boxes are routines external to matlab. Cylinders are databases.



Data Analysis Routines

Output types include graphical figure displays and text files.