Characterizing Cell Types through Differentially Expressed Gene Clusters Using a Model-Based Approach

Juliane Perner and Elena Zotenko

Max-Planck Institute für Informatik,
Computational Biology and Applied Algorithmics,
Campus E1 4,
66123 Saarbrücken, Germany
{jperner,ezotenko}@mpi-inf.mpg.de

Abstract. Expression profiles of all genes can aid in getting more insight into the biological foundation of observed phenotypes or in identifying marker genes for use in clinical practice. With the invention of high-throughput DNA Microarrays profiling the expression state of cells on a whole-genome scale became feasible.

Here, we propose a method based on model-based clustering to detect marker gene clusters that are most important in classifying different cell types. We show at the example of Acute Lymphoblastic Leukemia that these modules capture the expression state of different sample classes and that they give more biological insight into the different cell types than using just marker genes. Additionally, our method suggests groups of genes that can serve as clinical relevant markers.

Keywords: Marker Selection, Model-based Clustering, Gene Expression Analysis, Acute Lymphoblastic Leukemia.

1 Introduction

Even though the cells in an organism are based on the same genetic material various phenotypes are observed. Understanding how the genome is read in each cell is a central question in molecular biology. High-throughput DNA Microarrays made it possible to measure the global gene expression of an eukaryotic cell and thus reveal the cell-specific expression pattern and the regulatory programs that are active [1].

Using these patterns, Microarrays can serve as a diagnostic tool, for example, to distinguish normal from disease cells or to find subtypes of diseases, as for instance in the case of *Acute Lymphoblastic Leukemia* (ALL). ALL is an heterogeneous cancer of white blood cells. Patient's subtypes are based on the genetic lesion that is present in the cell and have different prognostic outcomes [2]. Thus classifying the patient's subtype is of great clinical value.

Microarray data is prone to various sources of noise that affects the measurements including cross-platform, laboratory-dependent or experimental noise.

P. Perner (Ed.): ICDM 2011, LNAI 6870, pp. 106–120, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Different normalization methods have been proposed to circumvent this problem. Still the impact of these methods on the actual results of statistical methods has been studied rarely.

Despite the potential drawback of noise, different studies [3,4,5] showed at the example of ALL that using Microarray data an accurate classification of disease subtypes is possible. Moreover, *marker genes* for which one can easily scan in clinical practice and that discriminate between the different classes based on their expression were detected.

A recent study [6] combined data of various sources to define a set of marker genes. These marker genes were sufficient to accurately classify a set of samples even from an independent experiment and distinct ethnic group. Further the authors analyzed the marker gene set to give biological insight into the disease by discovering enriched *KEGG* pathways. However, the interpretation of the results of the functional enrichment analysis might be hampered by the fact that marker gene sets contain solely the most differentially expressed genes but not necessarily functionally related genes. Thus, statistical methods applied in functional enrichment analysis might have problems in identifying enriched functional categories within the marker gene set. Further, clinical application of the marker genes might be hindered due to the lack of simple experimental procedures for detection.

Clustering methods have been used to detect groups of genes whose expression is similar across different samples [7]. The assumption is that genes that share the same expression pattern might be similar in their function. A cluster is thought to reflect a regulatory module of genes, that is switched on or off in concert if needed, within the cellular transcription network.

Detecting modules that discriminate the sample classes the most and using them to describe and analyze an observed class would provide more insight into the regulatory mechanisms underlying each class and more choices for selecting clinically relevant markers. In the following we call those modules *marker modules*.

In this paper, we introduce a method that is based on a model-based clustering, to detect marker modules. The expression values of genes in the gene clusters are summarized and a simple feature selection method using a Support Vector Machine (SVM) is applied to this data summary to learn the marker modules. Analyzing the marker modules we hope to find an enrichment of specific pathways or biological categories that give more insight into the biological foundation of the classes. We validate the concept of our approach at the example of ALL data collected from independent studies as in [6].

Model-based clustering has been applied to expression data from Microarray studies before (e.g. [8,9]). These methods discriminate in their model formulation for the specific tasks. Yeung et al. [8] attempt to cluster the genes without explicitly taking into account that the genes might be differently expressed according to the known sample classes. Segal et al. [9] try to cluster genes and samples at the same time and further learn the regulatory mechanism that could explain the observed two-way clustering. The method that is closest to our approach

is implemented in *PCluster* [10] which uses an heuristic procedure to detect a local optimal clustering of genes with a fixed partition of samples. Our model gives a better picture on the quality of a clustering since we are going to sample different local optima.

To the best of our knowledge none of the model-based clustering approaches has been used to detect marker modules. But the idea of summarizing gene expression data based on clusters and to use these clusters to train a classifier was applied before [11]. We apply a different learning procedure which learns the optimal level of detail of the clusters directly from the data.

The remainder of this paper is organized as follows. In section 2 we introduce the exemplary data and the pre-processing steps used in the analysis. Section 3 explains our model formulation and optimization method. The results are shown in section 4. The paper is concluded with a summary and discussion.

2 Material

The gene expression data of the ALL subtypes used in this paper was collected and pre-processed as described in [6]. It consists of four studies that were measured on various platforms (Affymetrix HU95a and HU133a GeneChips) and by different laboratories. The sample numbers per class and study are depicted in Tab. 1.

Additionally, we normalized the data set using different normalization methods. In log-2 transformation, each expression value was replaced with its logarithm. For Rank-normalization, the expression values within one sample were replaced by their normalized ranks such that they are uniformly distributed in the interval $[0, 1]$. In Z-normalization the expression values within one sample were scaled to have mean 0 and standard deviation 1. This resulted in three differently normalized data sets to which we applied our approach independently.

We maintained only genes that are differentially expressed across the ALL-subgroups to reduce the number of non-informative genes and to make the subtype-specific expression pattern more obvious. Differentially expressed genes

Table 1. Number of samples in each subtype (class) per original study. First column gives the name of the subtype.

	Ross et al. [4]	Hoffmann et al. [5]	Yeoh et al. [3]	Li et al. [6]
1 BCR-ABL	15	3	16	6
2 E2A-PBX1	18	3	27	7
3 Hyperdipl.>50	17	17	65	22
4 MLL	20	7	21	4
5 T-ALL	14	37	45	10
6 TEL-AML1	20	1	79	29
Total number	104	68	253	100

were detected by applying the *RankProd* method [12] to each subclass separately taking the remaining samples as reference. The 500 top up-regulated and 500 top down-regulated genes per class were chosen for further analysis. The lists for each subclass were merged and the overlap in these lists decreased the total numbers of retained gene to the following depending on the normalization method: Rank - 2310; Z-norm - 1657; log2 - 2244; unnormalized - 1550.

3 Methods

In order to detect and analyse a set of marker modules, we applied a 4 step process including a) pre-processing (described in section 2), b) gene clustering, c) marker module detection and d) biological analysis. These marker modules should be such that they are sufficient for distinguishing samples from different classes and genes within one module have similar expression patterns over the different classes.

3.1 Model-Based Clustering

The noisiness of the data suggests to use a Model-based Clustering approach where the expression values within the clusters are described by Gaussian Random Variables. Similarity between genes is defined by the likelihood of observing their expression values together under the assumption that they are random samples from the same Gaussian distribution. It is presumed that the whole data set was generated by a mixture of cluster-specific distributions. The objective is to find an assignment of genes g to a partition G_k such that the optimal set of partitions C^* is given by the set of partitions that maximizes the likelihood of the data.

We extend the problem formulation by introducing a dependency of the expression values on the known sample class S_l. To find the optimal partitions we want to maximize the likelihood

$$P(D|C,\theta) = \prod_{k=1}^{K} \prod_{l=1}^{L} \prod_{g \in G_k} \prod_{s \in S_l} p(e_{gs}|\mu_{kl}, \sigma_{kl}) \tag{1}$$

of the data given the clustering $C = \langle G, S \rangle$ and all the parameters θ of each Gaussian distribution. As the expression values are assumed to be independent given the clusters, the likelihood separates into local probabilities. The model formulation stresses the fact that genes within one gene cluster should have similar expression within one sample class but are allowed to have different expression between sample classes.

For optimization we consider the Bayesian score that marginalizes the effect of the parameters on the clustering and is defined as the logarithm of the posterior probability of a clustering. The Bayesian score thereby incorporates the uncertainty in the choice of the parameters by treating parameters and cluster assignments as Random Variables as well and averages the likelihood over all possible parameter choices.

The posterior of a clustering C given the data D is defined as

$$P(C|D) \propto P(D|C) * P(C)$$
$$= \int \int \prod_k \prod_l \left[p(\mu_{kl}, \sigma_{kl}) \prod_{g \in G_k} \prod_{s \in S_{(k,l)}} p(e_{gs}|\mu_{kl}, \sigma_{kl}) \right] d\mu_{kl} d\sigma_{kl} \quad (2)$$

up to a constant factor and under certain assumptions (for details see [13]). The Bayesian Score decomposes into a sum of cluster-specific scores. Further, evaluating the score of a cluster can be done easily by computing the *sufficient statistics* of the expression values in each cluster C_{kl} that summarizes the large amount of data by a minimum of values per cluster and is independent of the parameters of the cluster [14]. Note that we are not explicitly using a similarity metric but rather minimize the variance of the expression values within a cluster indirectly by using the sufficient statistics of the cluster and optimizing the closed form of the double integral in 2. Please refer to [14] or [10] for details.

3.2 Gibbs Sampler Algorithm for Cluster Optimization

To get a good picture of the posterior probability and to capture many solutions of high quality (i.e. local optima) the posterior probability can be sampled using the Gibbs sampler approach which is a stochastic approximation technique. At each sampling step just one variable assignment is changed and after *burn-in* the Gibbs Sampler is thought to generate clusterings from the posterior distribution.

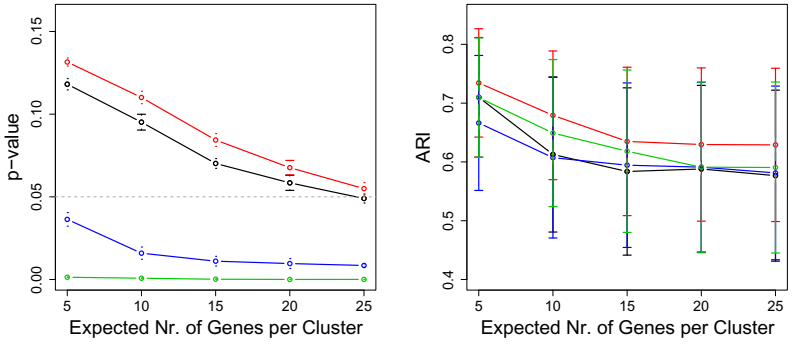
We adopt the Gibbs sampling procedure of Joshi et al. [14] but modify it to incorporate the fixed sample partitions during learning and a fixed total number of clusters K. The first modification is done because we know the sample classes in advance and want the clusters to be able to have distinct expression between these known classes. The second adjustment is introduced to get a better picture of how the method attempts to cluster the data.

The implementation of the Gibbs sampling starts with initializing the clustering by randomly distributing the genes across the K clusters. Afterwards, the following three steps are iterated until burn-in: First, a random gene i is selected. For this gene the difference in Bayesian score when assigning gene i to any other cluster while keeping all other assignments fixed is calculated. Third, the reassignment of gene i to a new cluster is accepted with a certain probability based on difference in Bayesian score. This procedure assures an iterative improvement of the clustering. For details on the implementation please refer to [14].

4 Results

In the scope of this paper we are addressing methodological and biological questions. We studied whether the expression values within the observed modules are following a normal distribution, what the impact of the different data normalization methods on our method is and how the number of modules affects the interpretation. The biological interpretation involves the detection of interesting KEGG pathways and their initial analysis.

We tested our approach under different conditions on the ALL data set described in section 2. Each experiment was performed on 5 different numbers of clusters K that correspond to an average of 5, 10, 15, 20 and 25 genes per cluster. Note that the quality of the clustering solutions largely depends on the ability of the Gibbs sampler to sample from the whole distribution [15]. We therefore performed 10 runs starting from different initial random clusterings for each experiment. The Gibbs sampler was run for 5000 iterations with a burn-in at iteration 1000. From the iterations after burn-in we used the clustering with highest score for further analysis.



(a) Lilliefors test for normality on modules

(b) Adjusted Rand Index between runs with different number of clusters

Fig. 1. (a) Average significance values of Lilliefors test for normality over decreasing number of clusters and for different normalization methods. (b) Average Adjusted Rand Index between the 10 runs with different number of clusters within one normalization method. Colours: black - Rank normalization; red - Log2 transformation; blue - Z-Normalization; green - No normalization.

4.1 Cluster Stability and Quality

Using the Lilliefors test for normality [16] we checked whether the normality assumption is fulfilled by the expression values in the clusters. P-values were computed for the expression values within each combination of gene cluster k and sample class c and then averaged per run. With a p-value below 0.05 the hypothesis that the observed population is sampled from a normal distribution is rejected. These p-values hint to how pure the clusters are within one Gibbs sampler run with respect to the normality assumption.

The adjusted Rand Index (ARI) [17] was used to give an idea of how similar the resulting clusters are over multiple Gibbs sampler runs or over results on the differently normalized data. We assume that we arrived at an optimal clustering if the ARI is close to 1 indicating that a lot of genes consistently cluster together.

Experiments in this section were performed on all 425 available training samples.

Number of Modules. Figure 1(a) shows the average p-values and their standard deviation over 10 different Gibbs sampler runs plotted with decreasing number of clusters K. The p-values for all normalization methods decrease when allowing fewer clusters showing that the method is forced to do an unfavourable union of genes into larger clusters or has to fit outliers into a cluster.

The average ARI comparing runs with the same number of clusters and same normalization method are depicted in Fig. 1(b). The highest ARI is achieved when using a large number of clusters and decreases when allowing less clusters but levels out at a value around 0.6 starting at 15 genes per module. The reason for this observation might be that until a number of 15 genes per module the method fits outliers into the clusters where there are a lot more possibilities to choose from than when it has to merge larger clusters together. This could make the clusterings more diverse.

If we compare runs of one cluster count to the runs of the next smaller cluster count we receive an average ARI of 0.4 - 0.56 which supports the idea that clusters are merged/splitted but genes are not completely shuffled.

Normalization. The effect of the various normalization methods on the clusters is also shown in Fig. 1(a). For Z-normalized and unnormalized data the p-values are below 0.05 and hence we would assume that the expression values in each module are not following a Gaussian distribution. In contrast, Rank normalized and log-transformed data could have been generated by a Gaussian distribution. The reason for this observation might be that for Z-normalization and unnormalized data the overall distribution of expression values in a sample is elongated towards high expression values and thus, it is difficult for the method to find a suitable cluster for the highly expressed genes. These genes might undergo only a small fold-change in expression in one sample but still will hardly fit the normal distribution derived from the expression values of other samples within the same sample class.

To check the consistency of the clusterings over differently normalized data we computed the adjusted Rand Index between two clusterings using only genes that are present in both datasets. Table 2 shows the average ARI over 10 runs between different normalization methods. Runs within one normalization method have an average ARI of approximately $0.66-0.73$ depending on the normalization method

Table 2. Average Adjusted Rand Index between two runs on differently normalized data. Agreement on clustering with on average 5 genes per cluster. Index was computed on genes that were present in both data sets.

	Rank	Log2	Z-norm	Unnorm
Rank	0.710	-	-	-
Log2	0.249	0.734	-	-
Z-norm	0.112	0.207	0.666	-
Unnorm	0.121	0.194	0.245	0.709

and are thus very similar. In contrast clusterings obtained with differently normalized data are quite distinct from each other although the Gibbs sampler runs start at the same initial clustering. There is some agreement with an adjusted Rand Index of approximately $0.11 - 0.28$. From that we deduce that there are certain genes that cluster tightly together over different normalization methods and build a core clustering while other genes are more variable.

Further, the clusterings between rank normalization and log2-transformation or unnormalized and z-normalized data seem to be more consistent than the other pairs. We assume that this is due to the similar effects of rank normalization and log2-transformation on the data, as well as the fact that z-normalization is just a scaled version of the unnormalized data.

4.2 Prediction Accuracy

For the detection of the marker modules we need to summarize the data within the gene clusters (in the following called *modules*). This is done by averaging the expression values of the genes falling within one of the K gene cluster per sample such that each sample is now described by the K average expression values.

In this section, we show that by doing so we do not destroy any structure in the data that is necessary to predict the sample class and hence show that the modules are valuable in explaining the classes. For this purpose, we validated our approach by calculating the 10-fold *Cross-validation* (CV) accuracy and the accuracy on the test set from [6] using a linear SVM. We also compared the performance using SVM to other classifiers but found that SVM performs best in terms of accuracy (data not shown) and the results are comparable to the approach of Li et al. [6] who used a marker gene selection method based on SVM.

For 10-fold CV we randomly distributed the available training samples into 10 sets of approximately the same size. We made sure that each set contained more than one sample from each of the six subclasses. Nine out of ten sets were used to train our model (modules and SVM) and the remaining set served for validation.

Table 3. CV and test accuracy all genes vs. modules over different normalization methods. Second column gives number of genes in total. "Clust." gives the best performing number of clusters. The mean accuracies and standard deviations of the SVM trained on all genes or modules are given in the column denoted with "Genes" or "Modules", respectively.

Norm.	#Genes	SVM - CV			SVM - Test		
		Clust.	Genes	Modules	Clust.	Genes	Modules
Rank	2310	462	99.05 (1.22)	99.09 (1.40)	231	100.00	98.21 (0.66)
Z-norm	1657	330	97.90 (2.54)	96.96 (2.57)	331	98.71	98.08 (0.68)
Log2	2244	224	99.07 (1.19)	98.67 (1.41)	448	98.71	100.00 (0.00)
Unnorm	1550	310	96.73 (3.15)	95.97 (3.12)	310	93.58	96.54 (1.22)

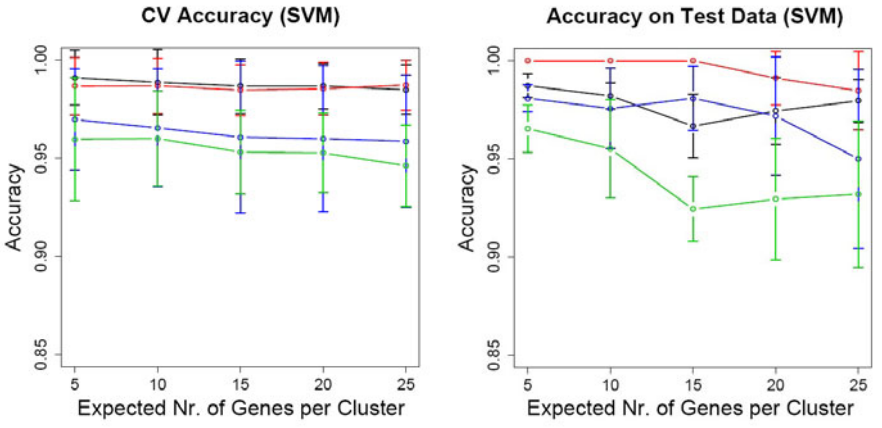


Fig. 2. (a) Average CV accuracy and standard deviation over the 10 CV-folds for different number of clusters. (b) Average test accuracy and standard deviation over multiple runs with different number of clusters. Colours: black - Rank normalization; red - Log2 transformation; blue - Z-Normalization; green - No normalization.

To test how the method can generalize to unseen data from a different laboratory and ethnic group we used all the 425 training samples for learning and afterwards classified the Li data set [6].

Normalization and Accuracy. In the first part of Tab. 3 the best performing average accuracy and standard deviation over the 10 CV-folds on the gene-by-gene data and the module summary are given for every normalization method. The CV results show that the classifier based on all genes gives slightly better result than the module-based approach (max. 1% accuracy loss) but all mean accuracies lie within standard deviation of the gene-by-gene approach. Based on CV accuracy we can not detect any severe difference between the normalization methods.

The results on the test set are presented in the second part of Tab. 3 showing the mean accuracy and standard deviation over the 10 runs of our method for the best performing number of clusters. Comparing the results of the gene-by-gene approach to our method, we find that SVM on modules outperforms the SVM using gene-by-gene data when using log2-transformed and unnormalized data. Our approach using SVM on log-transformed data classifies all test samples perfectly in all runs. We attribute the increase in performance on unnormalized data to the blurring of outliers by averaging the gene expression values in modules. The accuracy drops slightly (max. 2%) when using modules on rank-normalized or z-normalized data.

From the performance in the CV and on the test set we conclude that we are not loosing information when summarizing the expression data according to our modules.

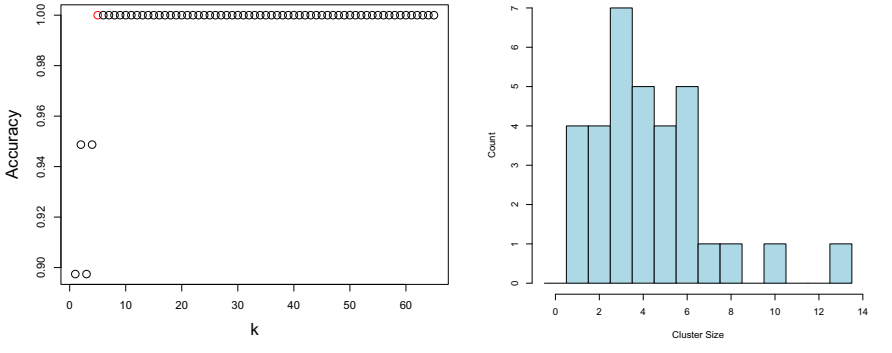
Number of Modules. Figure 2(a) and 2(b) show the prediction accuracies with an increasing number of clusters for the 10-fold CV and on the test set, respectively. The classification in the CV is stable over different cluster counts on log-transformed and rank normalized data (loss in accuracy less than 1%). We suspect that there is a hierarchy in the clusters. Thus genes that are fairly similar might be separated in small clusters when allowing a high cluster number but are clustered together when having fewer clusters to choose from. This hypothesis is also supported by the findings in section 4.1.

For further analysis we use the number of modules that perform best in terms of accuracy (Tab. 3) and cluster stability as we believe that they are the most appropriate clustering of genes and result in the best summary of the genes within the modules.

4.3 Marker Module Detection and Analysis

For the biological analysis of the modules we used one particular run and parameter setting based on the observation made in the last section. We analyze the first run on all of the available log-transformed training samples with 149 clusters (refers to an average of 15 genes per cluster). This setting performed best when looking at cluster purity and at the performance of the classifiers based on modules. We selected the lowest best performing cluster number to get meaningful groups of genes.

Marker Module Selection. Using the modules we summarized the expression data for each sample by averaging over the expression values of the genes in each module. For our particular run we have additionally checked the variance of the clusters per sample before averaging and found that the clusters have low variance (median variance per cluster and sample 0.005 after normalizing



(a) Number of marker modules (b) Marker module size distribution

Fig. 3. (a) Number of top-ranked modules per class pair for classification plotted against accuracy on test set. At $k = 63$ all modules are used. (b) Histogram of the number of genes in the 33 selected marker modules.

samples to variance 1). Thus the expression values define the valid regions of the clusters and we can assume that the mean of the expression value population per cluster resembles the true expected expression of the cluster.

Subsequently, this summary and the predefined class labels were used to train a SVM with linear kernel. From the feature coefficients of the SVM the clusters having highest impact on the classification were learned. We call this set of modules 'marker modules'.

In detail, to obtain the marker modules we performed a 10-fold CV on the training samples and collected the scaled coefficients for each one-vs.-one linear SVM. Next, for each classifier discriminating two classes, we averaged the coefficients per module over the different CV-folds and ranked the resulting average coefficients.

The k (where $k = 1, ..., 149$) top-ranked modules per pair of classes were selected to train a SVM on all training samples on the merged module set. The accuracy when classifying the test set is shown in Fig. 3(a) for increasing k. The figure shows that the accuracy stabilizes very quickly starting at $k = 5$.

Combining the 5 top-ranked clusters from each classifier we ended up with 33 marker modules that contained 1 to 13 genes (see Fig. 3(b)). All selected modules together contained 141 genes. The heatmap of the selected modules in Fig. 4 shows that the average expression of the modules clearly differs between the classes. Further, we found that the genes within the modules are sufficient to group the samples according to classes in an unsupervised fashion (data not shown).

Comparison to Marker Genes. Using a SVM on all genes we selected marker genes using the approach described above. These marker genes were compared to the genes in the marker modules. Out of the 27 marker genes detected with this procedure 17 genes were also present in the marker modules. These marker genes are distributed over 14 modules and can be seen as representatives of the genes within the marker modules.

Recently, Li et al. [6] also detected 62 marker genes using the same data set and an iterative feature reduction approach based on *SVM-RFE*. Only 43 of these are contained in our set of 2244 differentially expressed genes. The marker

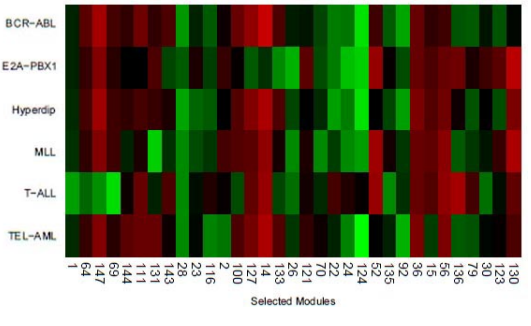


Fig. 4. Heatmap of the average expression values within the marker modules and classes. Colours range from red(no expression) to green(highly expressed).

modules overlap to this set in 16 genes that are distributed over 13 modules. Using their marker genes to classify the test data they achieved 99% accuracy which is comparable to our classification performance.

Functional Similarity of Genes in Marker Modules. The marker modules were further analyzed to detect enrichment for KEGG pathways. We used the GeneTrail Server [18] to perform an *Over-/Under-Representation analysis* (ORA) on the genes in the marker modules.

A selection of the most significant KEGG annotations per module is shown in Tab. 4. Among the detected KEGG pathways immune-system and haematopoietic linage related pathways are overrepresented. Overall 13 marker modules were found to have KEGG pathways enriched. Together we detected 24 KEGG pathways and overall 27 genes were responsible for the observed KEGG pathways. When performing ORA on all 141 genes in the modules 18 KEGG pathways were found. Only in a few cases these modules comprise more genes than when performing ORA on each module separately. Thus genes in one KEGG path seem to be condensed in one or two modules maximum.

We compared our results to the KEGG pathways detected in [6]. With their marker genes they detected 12 pathways out of which 6 were also detected with the marker modules. These 6 pathways comprise mostly immune system specific

Table 4. Exemplary KEGG pathways

KEGG	Description	Module	p-value	Genes detected
05320	Autoimmune thyroid disease	116	0.00006	HLA-DMB HLA-DQB1 HLA-DMA
04514	Cell adhesion molecules (CAMs)	116	0.00086	HLA-DMB HLA-DQB1 HLA-DMA
		121	0.00573	PECAM1 CD34
04512	ECM-receptor interaction	36	0.01828	COL6A3 LAMA3
04640	Hematopoietic cell lineage	92	0.00398	MME DNTT
		64	0.00778	CD3E CD2
04672	Intestinal immune network for IgA production	116	0.00006	HLA-DMB HLA-DQB1 HLA-DMA
04670	Leukocyte transendothelial migration	28	0.01804	MSN CD99
04650	Natural killer cell mediated cytotoxicity	100	0.00197	CD247 ZAP70
05340	Primary immunodeficiency	69	0.00087	LCK CD3D
04660	T cell receptor signalling pathway	69	0.00113	LCK CD3D
		100	0.00197	CD247 ZAP70
		64	0.00778	CD3E PRKCQ

pathways and signalling pathways. Interestingly, only 14 genes account for the KEGG pathways identified in [6] and most of the pathways are enriched because of the contribution of only 4 genes, e.g. the gene *PKI3R3* appears in 10 out of 12 pathways. Looking at our KEGG pathways, although 11 of the pathways are solely found due to module 116 and 124, the other pathways are found based on different modules. Thus we detect the pathways through the support of many different genes and in that way detect pathways that would not have been observed using marker genes.

5 Summary

We have proposed a method to detect differentially expressed marker modules in gene expression data across different cell types. The method is such that it first clusters the data using a model-based approach to find groups of gene that have a similar expression pattern across the sample classes. These modules are thought to capture a significant number of functionally related genes. Next, the data is summarized utilizing these modules and used to train a classifier from which the most important modules in discriminating the classes were deduced. These marker modules can then be further investigated for functional similarity of the genes within the modules.

We analyzed our method at the example of classifying subtypes of Acute Lymphoblastic Leukemia. In this process we answered different methodological question. We have illustrated that the clusterings are stable. Further, normalization seems to have a great impact onto the quality of the clusters in term of their purity and stability but not in terms of the performance of the classifier. Our finding shows that one has to choose carefully the normalization method for each experiment to get biological relevant results.

Moreover, we showed that using the marker module approach one can detect more significant functional annotations and hence get more biological insight into the subgroups of ALL. Our marker modules compares well to the results of a comparable marker gene approach or a recently published procedure [6]. We conclude that we are not loosing the marker gene information but rather extend the information by collecting more functional related genes for each marker gene.

Overall we conclude that marker modules are a promising approach to detect functional similarity of class-discriminating genes and thereby giving more biological insight into the underlying biological sources and regulatory mechanisms behind the observed phenotype of a cell. Further, marker modules provide a wealthy source for the selection of markers that can be applied in clinical practice.

Our approach is open for different modifications and extensions. First, other optimization methods, e.g. hill-climbing or simulated annealing techniques, for the model-based clustering could be tested. Second, the model formulation is flexible and one could incorporate other data sources that can aid the detection of modules. This could be known transcription binding sites or RNA-knockout scans. Third, the different clusterings resulting from several Gibbs Sampler runs

could be used to find core clusters that comprise genes that are constantly clustered together and thus might be more coherent in their function. Moreover, the marker module selection method is rather simple but easy to interpret. One could try other more advanced feature selection methods on the cost of interpretability. Finally, our method can be adopted for RNA-Sequencing data.

References

1. Bhattacharya, S., Mariani, T.J.: Array of hope: expression profiling identifies disease biomarkers and mechanism. Biochemical Society Transactions 037(4), 855–862 (2009)
2. Downing, J.R., Shannon, K.M.: Acute leukemia: A pediatric perspective. Cancer Cell 2(6), 437–445 (2002)
3. Yeoh, E., Ross, M.E., Shurtleff, S.A., Williams, W.K., Patel, D., Mahfouz, R., Behm, F.G., Raimondi, S.C., Relling, M.V., Patel, A., Cheng, C., Campana, D., Wilkins, D., Zhou, X., Li, J., Liu, H., Pui, C.H., Evans, W.E., Naeve, C., Wong, L., Downing, J.R.: Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell 1(2), 133–143 (2002)
4. Ross, M.E., Zhou, X., Song, G., Shurtleff, S.A., Girtman, K., Williams, W.K., Liu, H.C., Mahfouz, R., Raimondi, S.C., Lenny, N., Patel, A., Downing, J.R.: Classification of pediatric acute lymphoblastic leukemia by gene expression profiling. Blood 102(8), 2951–2959 (2003)
5. Hoffmann, K., Firth, M., Beesley, A., de Klerk, N., Kees, U.: Translating microarray data for diagnostic testing in childhood leukaemia. BMC Cancer 6(1), 229 (2006)
6. Li, Z., Zhang, W., Wu, M., Zhu, S., Gao, C.: Gene expression-based classification and regulatory networks of pediatric acute lymphoblastic leukemia. Blood 114(20), 4486–4493 (2009)
7. Kerr, G., Ruskin, H.J., Crane, M., Doolan, P.: Techniques for clustering gene expression data. Computers in Biology and Medicine 38(3), 283–293 (2008)
8. Yeung, K.Y., Fraley, C., Murua, A., Raftery, A.E., Ruzzo, W.L.: Model-based clustering and data transformations for gene expression data. Bioinformatics 17(10), 977–987 (2001)
9. Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., Friedman, N.: Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nature Genetics 34(2), 166–176 (2003)
10. Friedman, N.: Pcluster: Probabilistic agglomerative clustering of gene expression profiles. Technical Report, Hebrew University (2003)
11. Hastie, T., Tibshirani, R., Botstein, D., Brown, P.: Supervised harvesting of expression trees. Genome Biology 2(1), 1–12 (2001)
12. Hong, F., Breitling, R., McEntee, C.W., Wittner, B.S., Nemhauser, J.L., Chory, J.: RankProd: a bioconductor package for detecting differentially expressed genes in meta-analysis. Bioinformatics 22(22), 2825–2827 (2006)
13. DeGroot, M.H.: Optimal Statistical Decisions. John Wiley & Sons, Inc., Hoboken (2004)
14. Joshi, A., Van de Peer, Y., Michoel, T.: Analysis of a Gibbs sampler method for model-based clustering of gene expression data. Bioinformatics 24(2), 176–183 (2008)

15. Medvedovic, M., Yeung, K.Y., Bumgarner, R.E.: Bayesian mixture model based clustering of replicated microarray data. Bioinformatics 20(8), 1222–1232 (2004)
16. Lilliefors, H.: On the Kolmogorov-Smirnov test for normality with mean and variance unknown. J. Am. Stat. Ass. 62, 399–402 (1967)
17. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: ICML 2009: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1073–1080. ACM, Montreal (2009)
18. Keller, A., Backes, C., Al-Awadhi, M., Gerasch, A., Künzer, J., Kohlbacher, O., Kaufmann, M., Lenhof, H.P.: GeneTrailExpress: a web-based pipeline for the statistical evaluation of microarray experiments. BMC Bioinformatics 9(1), 552 (2008)