BIOINFORMATICS

Vol. 22 no. 14 2006, pages e507–e513
doi:10.1093/bioinformatics/btl214

Novel Unsupervised Feature Filtering of Biological Data

Roy Varshavsky[1,*], Assaf Gottlieb[2], Michal Linial[3] and David Horn[2]

[1]School of Computer Science and Engineering, The Hebrew University of Jerusalem 91904, Israel, [2]School of Physics and Astronomy, Tel Aviv University 69978, Israel and [3]Department of Biological Chemistry, Institute of Life Sciences, The Hebrew University of Jerusalem 91904, Israel

ABSTRACT

Motivation: Many methods have been developed for selecting small informative feature subsets in large noisy data. However, unsupervised methods are scarce. Examples are using the variance of data collected for each feature, or the projection of the feature on the first principal component. We propose a novel unsupervised criterion, based on SVD-entropy, selecting a feature according to its contribution to the entropy (CE) calculated on a leave-one-out basis. This can be implemented in four ways: simple ranking according to CE values (SR); forward selection by accumulating features according to which set produces highest entropy (FS1); forward selection by accumulating features through the choice of the best CE out of the remaining ones (FS2); backward elimination (BE) of features with the lowest CE.

Results: We apply our methods to different benchmarks. In each case we evaluate the success of clustering the data in the selected feature spaces, by measuring Jaccard scores with respect to known classifications. We demonstrate that feature filtering according to CE outperforms the variance method and gene-shaving. There are cases where the analysis, based on a small set of selected features, outperforms the best score reported when all information was used. Our method calls for an optimal size of the relevant feature set. This turns out to be just a few percents of the number of genes in the two Leukemia datasets that we have analyzed. Moreover, the most favored selected genes turn out to have significant GO enrichment in relevant cellular processes.

Abbreviations: Singular Value Decomposition (SVD), Principal Component Analysis (PCA), Quantum Clustering (QC), Gene Shaving (GS), Variance Selection (VS), Backward Elimination (BE)

Contact: royke@cs.huji.ac.il

Conflicts of Interest: not reported

1 INTRODUCTION

Feature selection is an important tool in many biological studies. Given the large complexity of biological data, e.g. the number of genes in a microarray experiment, one naturally looks for a small subset of features (e.g. small number of genes) that may explain the properties of the data that are being investigated. This type of motivation fits into the general scheme of **feature exploration**, i.e. searching for features because of their direct biological relevance to the problem. An alternative motivation is that of **preprocessing**: searching for a small set of features to simplify computational constraints, to allow for the handling of high throughput biological experiments, and to separate signal from noise. Practically, selection of a small set of genes is of ultimate importance when a small set of informative genes can be the basis for cancer diagnosis and a basis for development of gene associated therapy.

Preprocessing often involves some operation on feature-space in order to reduce the dimensionality of the data. This is referred to as **feature extraction,** e.g. restricting oneself to the first r principal components of a PCA routine. Note that superpositions of features appear in this example. Alternatively, in **feature selection** we limit ourselves to particular features of the original problem. This is the subject to be studied here. Let us refer to Guyon and Elissef (2003) for a comprehensive survey.

It is conventional to distinguish between **wrapper** and **filter** modes of the feature selection process. Wrapper methods contain a well-specified objective function, which should be optimized through the selection. The algorithmic process usually involves several iterations until a target or convergence is achieved. **Feature filtering** is a process of selecting features without referring back to the data classification or any other target function. Hence we find filtering as a more suitable process that may be applied in an **unsupervised** manner.

Unsupervised feature selection algorithms belong to the field of unsupervised learning. These algorithms are quite different from the major bulk of feature selection studies that are based on supervised methods (e.g., Guyon and Elissef, 2003, Liu and Wong, 2002), and compared to the latter are relatively overlooked. Unsupervised studies, unaided by objective functions, may be more difficult to carry out, nevertheless they convey several important theoretical advantages: they are unbiased, by neither the experimental expert nor by the data-analyst, can be preformed well when no prior knowledge is available, and they reduce the risk of overfitting (in contrast to supervised feature selection that may be unable to deal with a new class of data). The downside of the unsupervised approach is that it relies on some mathematical principle, like the one to be suggested in this study, and no guarantee is given that this principle is universally valid for all data. A common practice to resolve this quandary is to demonstrate the success of the method on various biological datasets and compare the results obtained by the method with external knowledge.

Existing methods of unsupervised feature filtering include ranking of features according to range or variance (e.g., Herrero, 2003, Guyon and Elissef, 2003), selection according to highest rank of the first principal component ('Gene shaving' of Hastie *et al*. 2000,

*To whom correspondence should be addressed.

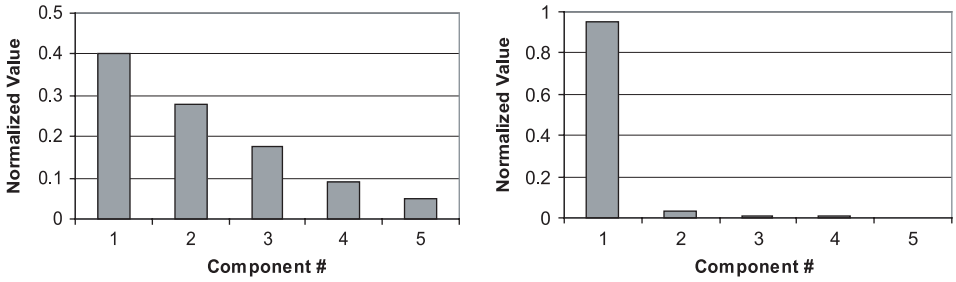

Fig. 1. A comparison of two eigenvalue distributions; the left has high entropy (0.87) and the right one has low entropy (0.14).

Ding 2003) and other statistical criteria. An example of the latter is Ben-Dor *et al*., (2001) where all possible partitions of the data are considered and the corresponding features are labeled. The partitions with statistical significant overabundance are selected. Another example is of Wolf *et al*., (2005), who optimize a function based on the spectral properties of the Laplacian of the features.

Here we present an intuitive, efficient and deterministic principle, leaning on authentic properties of the data, which serves as a reliable criterion for feature ranking. We demonstrate that this principle can be turned into efficient and successful feature selection methods. They compete favorably with other popular methods.

2 METHODS

2.1 Mathematical framework and notations

Let us consider a dataset of n instances[1] $A_{[nXm]} = \{\bar{A}_1, \bar{A}_{2,\ldots,} \bar{A}_{i,\ldots,} \bar{A}_n\}$, where each instance, or observation, $\bar{A}_i$ is a vector of m measurements or features. The objective is to define a subset of features $\tilde{M}$, of size $m_c < m$, that, in a sense to be defined below, best represents the data.

In PCA (or SVD) studies it is conventional to regard the best representation as the minimal least-square approximation of the original matrix (Wall *et al*., 2003). This principle can be followed also in feature extraction but it has the disadvantage that it may preserve too many properties of the data, including systematic noise. We will define our 'best approximation' using a principle based on SVD-entropy, and subject it to an a-posteriori test: given different selection rules of features choose the ones that prove useful as basis for the best fit to labeled data, e.g., perform clustering within the data-space spanned by the selected features and compare the results with known classification. This comparison will be performed using the Jaccard score.

$$J = \frac{n_{11}}{n_{11} + n_{01} + n_{10}} \tag{1}$$

where n_{11} is the number of pairs of instances that are classified together, both in the 'expert' classification and in the classification obtained by the algorithm; n_{10} is the number of pairs that are classified together in the 'expert' classification, but not in the algorithm's classification; n_{01} is the number of pairs that are classified together in the algorithm's classification, but not in the 'expert' classification.

The Jaccard score reflects the 'intersection over union' between the algorithm's clustering assignments and the expected classification. Its values range from 0 (no match) to 1 (perfect match).

2.2 Ranking by SVD-Entropy

Alter *et al*., (2000) have defined an SVD-based entropy of the dataset. Denote by s_j the singular values of the matrix A. s_j^2 are then the eigenvalues of the nxn matrix AA^t. Let us define the normalized relative values (Wall *et al*., 2003): and the resulting

$$V_j = s_j^2 \Big/ \sum_k s_k^2 \tag{2}$$

dataset entropy (Alter *et al*., 2000):

$$E = -\frac{1}{\log(N)} \sum_{j=1}^{N} V_j \log(V_j) \tag{3}$$

This entropy varies between 0 and 1. E = 0 corresponds to an ultra-ordered dataset that can be explained by a single eigenvector (problem of rank 1), and E = 1 stands for a disordered matrix in which the spectrum is uniformly distributed. Figure 1 demonstrates two examples of 5 eigenvalues, one with high entropy (left, 0.87) and the other with low entropy (right, 0.14). As can be seen in Figure 1, when the entropy is very low, one expects a very non-uniform behavior of eigenvalues. One should not confuse the standard definition of entropy, based on probabilities (Shannon, 1948), with the one used here, which is based on the distribution of eigen- (or singular) values. Although standard entropy considerations appear in feature selection methods, such as the supervised bottleneck approach (Tishby *et al*., 2000), the use of SVD-entropy for feature selection is a novel approach.

We define the contribution of the *i-th* feature to the entropy (CE_i) by a leave-one-out comparison according to

$$CE_i = E(A_{[nXm]}) - E(A_{[nX(m-1)]}) \tag{4}$$

where, in the last matrix, the *i-th* feature was removed.

Thus we can sort features by their relative contribution to the entropy. Let us define the average of all CE to be c and their standard deviation to be d. We distinguish then between three groups of features:

(1) $CE_i > c + d$, features with high contribution

(2) $c + d > CE_i > c\text{-}d$ features with average contribution

(3) $CE_i < c\text{-}d$ features with low (usually negative) contribution

Features in the first group (high CE) lead to entropy increase; hence they are assumed to be very relevant to our problem. Retaining these features we expect the instances to be more evenly spread in the truncated SVD space. The features of the second group are neutral. Their presence or absence does not change the entropy of the dataset and hence they can be filtered out without much information loss. The third group includes features that reduce the total SVD-entropy (usually $c\text{-}d < 0$). Such features may be expected to contribute uniformly to the different instances, and may just as well be filtered out from the analysis.

The first feature selection method that we propose is to limit oneself to the first group of features according to the CE ranking. A will then be represented by a new matrix of rank m_c, the number of features in group 1. Several other feature selection methods are suggested in the next section. In all of them we assume that the same value of m_c continues to serve as the right guide for optimal dimensionality reduction.

2.3 Three Feature Selection Methods

Entropy maximization can be implemented in three different ways, as is also the case in other feature selection methods.

[1]In this paper A (or $A_{[nXm]}$) is a matrix and $\overline{A}$ (or $\overline{A_i}$) is a vector.

1. Start with $\tilde{M} = \varnothing$ and $M' = M$
2. Select the element with the **highest** CE. Remove it from M', insert it into $\tilde{M}$
3. While size of $\tilde{M} < m_c$
   a. For each element in M' ($\forall m \notin \tilde{M}$) compute its CE score on $\tilde{M}$ ($E(A_{\tilde{M}+i}) - E(A_{\tilde{M}})$)
   b. Select the element with the **highest** CE Score → remove from M', insert into $\tilde{M}$
4. End

Box 1: Pseudo-code of Forward Selection method FS1

1. Start with $\tilde{M} = \varnothing$ and $M' = M$
2. While size of $\tilde{M} < m_c$
   a. Select the element in M' ($\forall m \notin \tilde{M}$) with the **highest** CE Score
   b. Remove from M', insert into $\tilde{M}$
3. End

Box 2: Pseudo-code of Forward Selection in method FS2

1. Start with $\tilde{M} = M$ and $M' = \varnothing$
2. While size of $\tilde{M} > m_c$
   a. Select the element in $\tilde{M}$ with the **lowest** CE Score
   b. Remove from $\tilde{M}$, insert into M'
3. End

Box 3: Pseudo-code of Backward Elimination method BE

(1) Simple ranking (SR): select m_c features according to the highest ranking order of their CE values.

(2) Forward Selection (FS): here we consider two implementations.

   (a) FS1: Choose the first feature according to the highest CE. Choose among all other features the one which, together with the first feature, produces a 2-feature set with highest entropy. Continue with iteration over all *m-2* features to choose the third according to maximal entropy, etc, until m_c features are selected (Box 1).

   (b) FS2: Choose the first feature as before. Recalculate the *CE* values of the remaining set of size *m-1* and select the second feature according to the highest *CE* value. Continue the same way until m_c features are selected (Box 2).

(3) Backward Elimination (BE): Eliminate the feature with the lowest CE value. Recalculate the CE values and iteratively eliminate the lowest one until mc features remain (Box 3).

One may view the different methods also as specifying alternative ranking methods. Whereas SR ranks the features according to their original CE values, FS1, FS2 and BE introduce other ranking orders through the algorithms defined above. In the examples studied below we display rankings for the entire range of 1 to *m*.

In an appendix we analyze the computational complexity of all these methods. SR is the fastest one and BE is the most cumbersome one for large numbers of features. In the examples to be discussed next, we will compare the different methods with one another. However, because of complexity, the BE method will be used in only one of the examples.

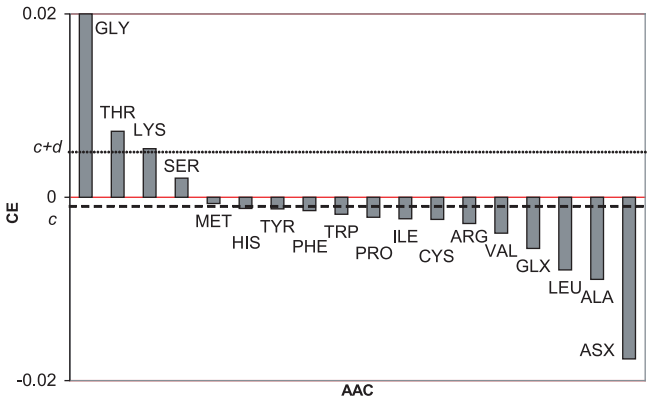

Fig. 2. CE of the 18 Amino Acid Compositions (AAC) of the virus dataset. ASX stands for ASN and ASP and GLX for GLN and GLU. The dashed line represents the value of *c* and the dot-dashed line the value of *c+d*.

3 Results

Our four feature filtering methods were compared with each other and with two known methods: Variance Selection (VS) and Gene Shaving (GS). The latter is a variation of a method of Hastie *et al*. (2000) which removes features iteratively according to their lowest correlations with the first principal component. For comparison we also look at results of random feature selection on several benchmarks.

3.1 The viruses dataset of Fauquet, 1988

This is a dataset of 61 rod-shaped viruses affecting various crops (tobacco, tomato, cucumber and others) originally described by Fauquet *et al*. (1988) and analyzed more thoroughly by Ripley (1996). There are 18 measurements of Amino Acid Compositions (AAC) for the coat proteins of the virus that serve as 18 features. The viruses are known to be classified into four classes: Hordeviruses (3), Tobraviruses (6), Tobamoviruses (39) and Furoviruses (13).

Figure 2 displays the CE values of all 18 features. Our criterion sets $m_c = 3$. We test the performance of the system for the entire *m* range to see if this choice makes sense. Before doing so, let us display the ranking orders of all methods in Table 1. By definition, SR has the same ranking order as CE in Figure 2. In this problem, BE turns out to lead to the same order as FS1, and all our three methods agree with each other on the first three features to be selected. We include in Table 1 also the ranking order of VS (variance selection) and GS (gene shaving). The two last ones are highly correlated with each other (Spearman correlation 0.76) but highly uncorrelated with our three methods (see Supplementary Material for more details). In particular note that VS chooses ASX and GLX as its second and third features, whereas for our three methods these two features are unfavorable (15^{th} to 18^{th}) choices.

Next we evaluate the subset selection using the Jaccard score. This is done by applying the QC clustering algorithm (Horn and Gottlieb, 2002) on the 61 viruses described by the selected subset of features. QC was applied after reduction of each space to normalized 3-space dimensions, using the parameter $\sigma = 0.5$ (for details see Varshavsky *et al*., 2005, and COMPACT[2]). Results are shown in

[2]http://adios.tau.ac.il/compact or http://www.protonet.cs.huji.ac.il/compact

Table 1. Ranking of the 18 Amino Acid Compositions of the virus dataset according to various feature filtering methods. Colors from white to black match the numbers that reflect the ranking of each method

AAC	SR	FS1/BE	FS2	VS	GS
GLY	1	1	1	1	9
THR	2	2	2	6	6
LYS	3	3	3	4	14
SER	4	13	4	5	4
MET	5	4	15	16	17
HIS	6	6	7	15	16
TYR	7	8	13	13	13
PHE	8	7	5	14	11
TRP	9	5	16	17	15
PRO	10	11	6	11	10
ILE	11	10	11	12	12
CYS	12	9	18	18	18
ARG	13	12	10	8	8
VAL	14	14	8	9	7
GLX	15	16	9	3	2
LEU	16	15	14	10	5
ALA	17	17	12	7	3
ASX	18	18	17	2	1

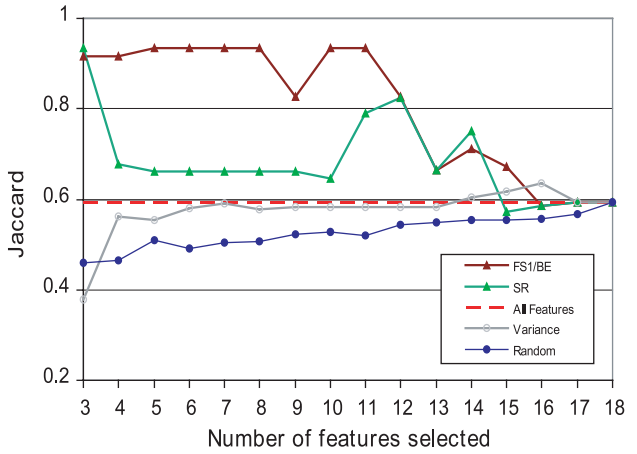

Fig. 3. Filtering quality of the virus dataset is tested by Jaccard scores of clustering performed in spaces spanned by them (see text). Best results are obtained for FS1 (identical with BE in this case) and SR for $m_c = 3$. FS1 continues to perform very well with more features. Feature selection according to VS performs worse. For comparison we include also an evaluation based on a large group of random order rankings.

Figure 3 for three of our four methods. All three do exceedingly well at the three features level (J > 0.9) whereas the variance method obtains J = 0.4. Note that our methods, with our choice of m_c, lead to a much better result than J = 0.6, obtained when all 18 features are taken into account. This exemplifies the importance of keeping features that maximize the entropy. The feature ranking of FS1 and BE is the only one that keeps performing very well with more than three selected features. Similar relative successes of feature selection evaluation (although less favorable J-scores) were obtained with other clustering methods, such as K-means. This comparison, as well as other details that could

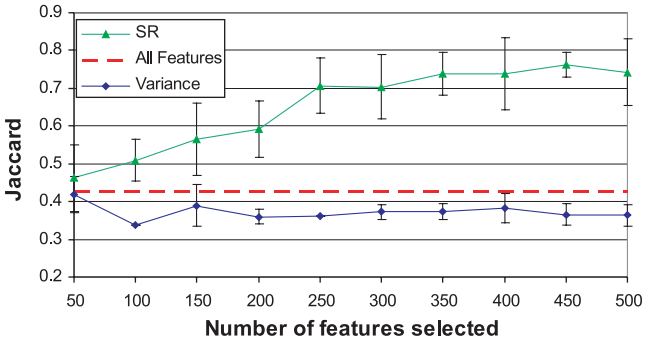

Fig. 4. Clustering quality of two feature selection methods. Results are averages of 100 runs of K-Means clustering.

not be fitted into this paper, can be found in the Supplementary Material[3].

Fauquet *et al*. (1987) have argued that the AAC of the coat protein of plant viruses are specific to the structure of the viral particle, to the mode of transmission and to sub-grouping of viruses to distinctive classes. Our results indicate that choosing only 3–4 features correctly, not only preserves the classification but allows much better performance with minimal failure. It is interesting to note that the 3 highest-ranking amino acids, GLY, THR and LYS are not dominating the coat proteins. These amino acids account for only 13–21.5% of the coat proteins, a fraction that is similar to the average percentage in the entire proteins database (18.3%). Further investigation shows that neither their size nor polarity or electric charges differentiate these three amino acids from the remaining. Nevertheless, since GLY, THR, LYS and MET (the fourth ranked AAC, according to the FS1 method) represent different functional groups, we conclude that the FS1/BE ranking is consistent with selecting amino acids that carry different physico-chemical properties.

3.2 The MLL dataset of Armstrong *et al*., 2002

The second dataset that we apply our methods to is that of Armstrong *et al*., 2002, who have attempted to cluster data of three Leukemia classes: lymphoblastic Leukemia with MLL translocations and conventional acute lymphoblastic (ALL) and acute myelogenous Leukemias (AML). In the experiment, 12582 gene expressions were recorded, using Affymetrix U95A chips on 72 patients, 20 of which diagnosed as MLL, 24 ALL and 28 AML. They showed that these 3 Leukemia types can be divided according to some gene expression. However, when filtering in an unsupervised manner (selecting 8700 genes that show some variability in expression level), the clustering results were unsatisfactory and much inferior to a supervised selection of 500 genes that best separate between the cancer patients.

Applying our CE criteria we use the method SR, and compare clustering of these feature-filtered data with VS (Figure 4). Clustering was performed by K-Means, averaging over 100 runs and using K = 3 with data projected onto a unit sphere in 3D-reduced space (Varshavsky *et al*., 2005). The asymptotic Jaccard score is J = 0.426 for this K-Means method. As can be seen in Figure 4 VS provides no improved quality, whereas SR leads to J-values

[3]http://adios.tau.ac.il/compact/UFF/SUPP

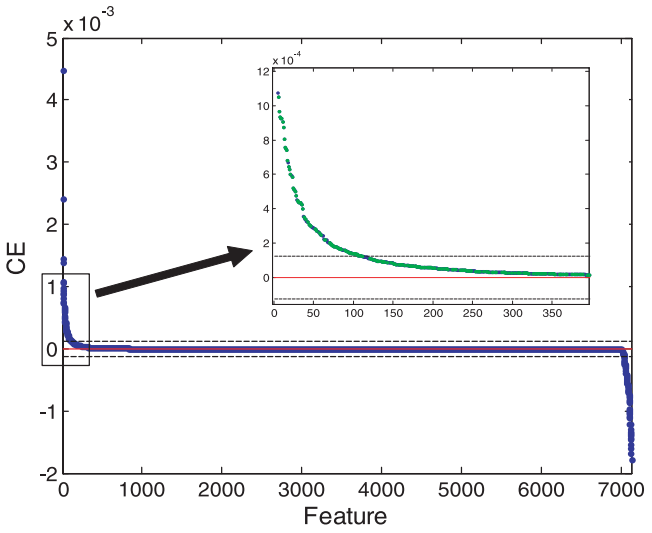

Fig. 5. CE of the 7129 genes of the Golub dataset ($c = 0$, dashed lines represent $c \pm d$). The inset zooms into the highest-ranked 300 genes, with bright dots signifying the top 100 features according to the FS1 method

between 0.7 and 0.8 for filtered gene groups of sizes 250 to 450. The preferred m_c value according to $c + d$ of SR is 254. Better results can be obtained by using the QC algorithm, but the same trend and conclusions regarding feature selection hold also there. It is interesting to note that QC clustering of our unsupervised SR method, for $m_c = 254$, reaches J = 0.85 (see supplementary).

We display the K-Means analysis in Figure 4, in spite of its poorer performance compared to QC, in order to emphasize that the quality of the feature filtering method is independent of the clustering-test performed on the filtered data.

3.3 The Leukemia dataset of Golub *et al.*, 1999

After demonstrating the effectiveness of our methods on both small and large datasets, we choose a third dataset (Golub *et al.*, 1999) that has served as a benchmark for several clustering algorithms (Sharan and Shamir, 2000, Getz *et al.*, 2000 and more) and feature selection methods (e.g., Liu B. *et al.*, 2004, Liu H. *et al.*, 2002). The experiment sampled 72 Leukemia patients with two types of Leukemia, ALL and AML. The ALL set is further divided into T-cell Leukemia and B-cell Leukemia and the AML set is divided into patients who have undergone treatment and those who did not. For each patient, an Affymetrix GeneChip measured the expression of 7129 genes. The task is clustering into the four correct groups within the 72 patients in a [7129x72] gene-expression matrix. This clustering task is quite difficult. Using the QC method (in normalized 5 dimensions with $\sigma = 0.54$), applied to the data without feature selection, one obtains J = 0.707, which is the best score for a variety of clustering algorithms (Varshavsky *et al.*, 2005).

The CE values for the 7129 features of this problem are displayed in Figure 5. Most of the features have a zero score. There are about 150 large CE values (see Figure 5) and about the same number of small CE values. The bright color within the inset indicates the first 100 features selected by FS1. While their ordering is different from the SR ranking, most of them belong, as expected, to the class of large CE values. The overlaps of the first leading features of SR

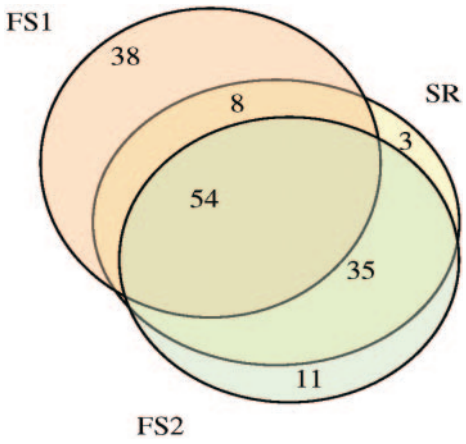

Fig. 6. Venn diagram of relations among the first 100 features selected by different methods.

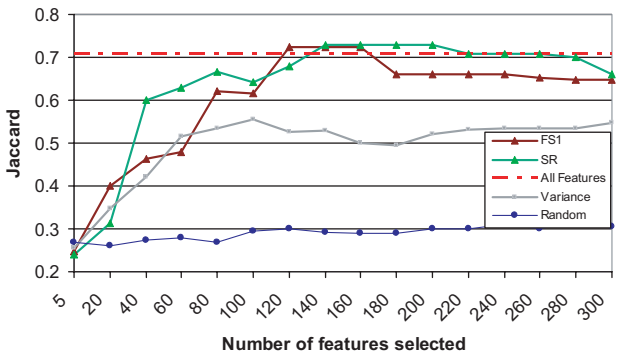

Fig. 7. Jaccard scores of QC clustering for different feature filtering methods on small gene subsets of the Golub data.

with those of FS1 and FS2 are shown in the Venn diagrams of Figure 6.

Next we turn to testing the filtering methods to see how well they do in the clustering task, i.e. what are the Jaccard scores that are obtained by applying an identical clustering algorithm to the different spaces spanned by the selected features. The clustering algorithm is the QC method mentioned above. Figure 7 shows that good results can be obtained by our filtering methods once the gene subset is larger than 100 or so. For feature sets of sizes 120 to 200 we find selections (of FS1 and SR) that lead to Jaccard scores that are better than J = 0.707, the asymptotic limit. Gene subsets larger than 300 result in Jaccard scores below the asymptotic limit (for a complete list, see supplementary). Also in this problem the GS results are inferior to those of the other methods.

3.3.1 Biological interpretations of the Leukemia dataset of Golub et al., 1999 It is clearly of interest to look at the 100 or so genes that participate in the sections that lead to the best Jaccard score. In Figure 6 we saw that there exists a substantial overlap between the choices of our three different methods. To study the biological significance of our subset of overlapping 54 genes we have run a GO enrichment analysis (NetAffx™ web tool[4]) on this subset. As

[4]http://www.affymetrix.com/analysis/index.affx

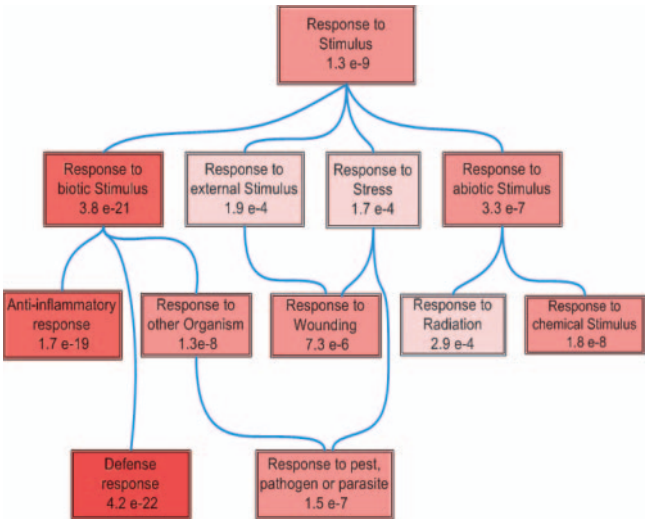

Fig. 8. Diacyclic graph of GO enrichment. Shown are GO nodes (Camon *et al*, 2004) with significant p-value of enrichment as determined by the NetAffx™ tool[4] (p-value < 5e-4). The color of each node matches its significance level (along the spectrum of red shades, light: lowest to dark: highest).

displayed in Figure 8 (and supplementary), we are able to assign some prevalent biological processes to the selected genes.

The association of our selected 54 genes with functional annotation related to defense, inflammation and response to pathogen (with p-value ranging from e-7 to e-22) is intriguing (Figure 8). It may underlie the difference in AML and ALL in view of the different susceptibility of the patients to treatment such as chemo and radiotherapy. Thus the listed protein processes may not only be considered as 'subtype cancer markers' but as an indication of the biological properties of the cancerous cells. Specifically, cellular response to pathogen, to stress and to inflammation may be different for AML and ALL. It may also provide a focused hypothesis towards the processes and mechanisms that can be used as a follow up in monitoring the outcome of therapy in case of Lymphoma.

4 Discussion

We have introduced a novel principle for unsupervised feature filtering that is based on maximization of SVD-entropy. The features can be ranked according to their CE-values. We have proposed four methods based on this principle and have tested their usefulness on three different biological benchmarks. Our methods outperform other conventional unsupervised filtering methods. This is clearly brought out by the examples that we have analyzed. More details are provided by our Supplementary Material[5]. In particular, it is striking to note how much more successful our methods are compared to VS, the popular variance ordered method.

The major theoretical difference between the two approaches is that VS relies on a measurement of one feature at a time. The entropy-based approach, as implemented by the CE calculation, takes into account the interplay of all features. In other words, the contribution of a feature, its CE, depends on the behavior of all other features in the problem. Thus variance is only one of the factors that affect the CE value. The CE value depends also on the correlations (or the absence thereof) of a given feature with all others. The difference between the ranking of SR and VS in Table 1 bears evidence to the difference between the two methods.

We have demonstrated that our selected features have important biological significance, through a GO enrichment analysis of the genes in the Golub dataset. A similar analysis of the Armstrong dataset is presented in the Supplementary Material[5]. In the virus dataset, we have shown that the FS1/BE filtering method works exceedingly well for a large range of numbers of features. The biological significance of the relevant choices of amino-acids remains to be uncovered.

The CE ranking leads to an estimate of the optimal m_c choice. This is an important point by itself. In other methods, such as VS, it is almost impossible to make this choice on the basis of variation of feature properties. Conventionally one makes therefore an arbitrary choice, such as selecting 10% or 50% of the features. In the three datasets discussed in our paper it seems quite clear that our suggested optimal m_c, as judged from the CE scores, leads indeed to optimal results. The improved Jaccard scores indicate that the selected m_c features have biological significance.

Our four methods differ in computational complexity. SR is the simplest one, since it relies just on sorting the initial CE values. In an appendix we compare its complexity with that of the other methods. The relative values depend on the choice of m_c (the size of the subset).

FS1 chooses features that lie high on the original CE-score, hence its optimal selected set will have a large intersection with that of SR. Nonetheless, for small numbers of selected features, the order may be very important. Thus, in the virus problem, FS1 turns out to be much more successful than SR. In the Leukemia datasets, where reasonable results were obtained for larger feature sets, FS1 was not found to be significantly better than SR. Biologically one may expect the appearance of features that are degenerate with one another, i.e. have quite identical behavior on all instances. Such duplicity can be included by the SR method but excluded by the FS1 one.

Our optimal feature-filtered sets in the two Leukemia problems turn out to include just few percents of all genes. Thus a CE-analysis indicates that a small subgroup of all genes is the most relevant one to the data in question. We have seen that this relevance is borne out by both Jaccard scores and GO enrichment analysis. The pursuit of small feature sets is often guided by wishful thinking that the essence of biological importance can be reduced to a small causal set. Here we find that the small number obtained in our analysis is an emerging phenomenon, and may be regarded as a true biological result.

ACKNOWLEDGEMENTS

We thank Alon Kaufman and Nati Linial for stimulating discussions and suggestions, and Orly Alter for technical and theoretical assistance. R.V. is supported by SCCB, the Sudarsky Center for Computational Biology in the Hebrew University of Jerusalem. This study was supported by the EU Framework VI NoE

[5]http://adios.tau.ac.il/compact/UFF/SUPP

DIAMONDS consortium, and also partially supported by the Israel Science Foundation (grant No. 39/02).

REFERENCES

Alter, O., Brown, P.O. and Botstein, D. (2000) Singular value decomposition for genome-wide expression data processing and modeling, PNAS, 97, 10101–10106.

Armstrong, S.A., Staunton, J.E., Silverman, L.B., Pieters, R., den Boer, M.L., Minden, M.D., Sallan, S.E., Lander, E.S., Golub, T.R. and Korsmeyer, S.J. (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia, 30, 41–47.

Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, LAPACK User's Guide (http://www.netlib.org/lapack/lug/lapack_lug.html), Third Edition, SIAM, Philadelphia, 1999.

Ben-Dor, A., Friedman, N. and Yakhini, Z. (2001) Class discovery in gene expression data. RECOMB. 31–38.

Camon E, Barrell D, Lee V, Dimmer E. and Apweiler R. (2003) Gene Ontology Annotation Database—An integrated resource of GO annotations to UniProt Knowledgebase. In Silico Biol., 4: 0002.

Ding, C., He, X., Zha, H. and Simon, H. (2002) Adaptive dimension reduction for clustering high dimensional data. IEEE International Conference on Data Mining. 107–114.

Ding, C.H.Q. (2003) Unsupervised Feature Selection Via Two-way Ordering in Gene Expression Analysis, Bioinformatics, 19, 1259–1266.

Fauquet, C., Desbois, D., Fargette, D. and Vidal, G. (1988) Classification of furoviruses based on the amino acid composition of their coat proteins. In Cooper, J.I. and Asher, M.J.C. (eds), Viruses with Fungal Vectors. Association of Applied Biologists, Edinburgh, 19–38.

Fauquet, C., Thouvenel, J. C. (1987). Viral diseases of plants in Ivory Cost. Intuition et Documentation Technique, 46. ORSTOM, Paris, 243 pp.

Getz, G., Levine, E. and Domany, E. (2000) Coupled two-way clustering analysis of gene microarray data, PNAS, 97, 12079–12084.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. and Lander, E.S. (1999) Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, Science, 286, 531–537.

Guyon, I. and Elisseeff, A. (2003) An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, 3, 1157–1182.

Hastie, T., Tibshirani, R., Eisen, M., Alizadeh, A., Levy, R., Staudt, L., Chan, W., Botstein, D. and Brown, P. (2000) 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns, Genome Biology, 1.

Herrero, J., Diaz-Uriarte, R. and Dopazo, J. (2003) Gene expression data preprocessing, Bioinformatics, 19, 655–656.

Horn, D. and Axel, I. (2003) Novel clustering algorithm for microarray expression data in a truncated SVD space, Bioinformatics, 19, 1110–1115.

Horn, D. and Gottlieb, A. (2002) Algorithm for data clustering in pattern recognition problems based on quantum mechanics, Physical Review Letters, 88.

Liu, B., Cui, Q., Jiang, T. and Ma, S. (2004) A combinational feature selection and ensemble neural network method for classification of gene expression data, BMC Bioinformatics,5. 136

Liu, H., Li, J. and Wong, L. (2002) A Comparative Study on Feature Selection and Classification Methods Using Gene Expression Profiles and Proteomic Patterns. In R. Lathrop, K.N., S. Miyano, T. Takagi, and M. Kanehisa (ed), 13th International Conference on Genome Informatics. Universal Academy Press, Tokyo Japan, 51–60.

Ripley, B.D. (1996) Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge.

Shannon, C. (1948) A mathematical theory of communication,, The Bell system technical journal, 27, 379–423, 623–656.

Sharan, R. and Shamir, R. (2000) CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis. AAAI Press, Menlo Park, CA, 307--316.

Sondberg-Madsen, N., Thomsen, C. and Pena, J.M. (2003) Unsupervised Feature Subset Selection. Workshop on Probabilistic Graphical Models for Classification. 71–82.

Tishby, N., Pereira, F., C. and Bialek, W. (2000) The information bottleneck method, CoRR, physics/0004057

Varshavsky, R., Linial, M. and Horn, D. (2005) COMPACT: A Comparative Package for Clustering Assessment. Lecture Notes in Computer Science. Volume 3759, 159–167. Springer-Verlag.

Wall, M., Rechtsteiner, A. and Rocha, L. (2003) Singular Value Decomposition and Principal Component Analysis. In Berrar, D., Dubitzky, W. and Granzow, M. (eds), A Practical Approach to Microarray Data Analysis. Kluwer, 91–109.

Wolf, L. and Shashua, A. (2005) Feature Selection for Unsupervised and Supervised Inference: The Emergence of Sparsity in a Weight-Based Approach, Journal of Machine Learning Research, 6, 1855--1887.

Xing, E.P. and Karp, R.M. (2001) CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. Bioinformatics, 17, S306–315.

APPENDIX

Computational complexity of the four methods

In the following calculations, we will assume that $m_c < n$, which will give upper bound to the complexity. We will not assume that $m < n$.

The computation of all eigenvalues for a dense symmetric matrix requires $O(p^3)$ operations, where p is the size of the matrix (Anderson, 1999).

We will define the complexity of the initial computation of all CEs to be $O(m^*min(n,m)^3) \equiv K$.

- SR: The computational complexity is lowest for the SR method. There's only one calculation of all CEs, followed by sorting. Hence the complexity is $O(K + m^*logm)$.
- FS1: Calculation of all CEs followed by $(m_c - 1)$ repetitive diagonalization of a growing matrix (from 2 to $(m_c - 1)$), leading to $O(K + m_*m_c^4)$.
- FS2: Calculation of all CEs followed by $(m_c - 1)$ repetitive diagonalization of a decreasing matrix (from m-2 to $(m\text{-}m_c)$), leading to $O(m^5\text{-}(m\text{-}m_c)^5)$. Note that here, if $n < (m\text{-}m_c)$, the complexity is $O(mm_cn^3)$
- BE: Calculation of all CEs followed by $(m\text{-}m_c\text{-}1)$ repetitive diagonalization of a decreasing matrix (from m-2 to $(m_c\text{-}1)$), leading to $O(m^5\text{-}m_c^5)$. Note that here, if n < m, the complexity is reduced to $O((m^2\text{-}m_c^2)n^3)$.

Clearly computational complexity is lowest for the SR method, since only one calculation of all CEs is needed. BE or FS2 have the highest complexity, depending on whether $m > 2m_c$ or not.