Mining large-scale Genomic and Proteomic Data: Algorithms, Tools and Inference

Thesis for the degree of

DOCTOR OF PHILOSOPHY

by

Roy Varshavsky

SUBMITTED TO THE SENATE OF

THE HEBREW UNIVERSITY OF JERUSALEM

December 2007

This work was carried out under the supervision of Prof. Michal Linial and Prof. David Horn

Acknowledgements

TBD

Abstract

The present era is a time of “genomic revolution”. In recent years, dozens of genomes have been sequenced, proteins and genes have been mapped and learned, their structures have been inferred and their functions are being understood.

The groundwork for such a rapid progress consists of several breakthroughs in high-throughput technologies that allow fast sequencing (e.g., Haplotype Map), recording expressions of thousands of genes simultaneously (e.g., Microarray chips, Comparative Genomic Hybridization), protein-DNA interactions (ChIP-on-chip), proteome properties (e.g., Mass Spectrometry, proteins chips) and more. Such technologies, accelerated by commercial platforms, have made data collection easy, reliable, relatively cheap and fast.

In conjunction with data-collection feasibility, progresses in storage and information transfer have facilitated the reposition and retrieval of that data. As a result, numerous online genomic repositories are now publicly available (e.g., NCBI, Stanford Genomics, Ensemble and UniProt).

This extensive availability of data has opened an opportunity for novel research directions, many of which were considered futuristic only a few years ago.Most of these efforts belong to a new discipline, strongly related to Bioinformatics, called “Systems Biology”. Systems Biology studies aim to determine the inter-relations among molecules (genes, RNA, metabolites and proteins), how groups of elements influence biological phenomena and how environmental and metabolic factors shape the cell’s ecosystem.

The first step to address these ambitious questions is to reveal hidden patterns out of the ‘clouds of data’. Methods that aim to extract information out of large-scale data are referred to as data mining techniques, and usually include statistical and machine learning principles. Existing Data Mining algorithms applied in Bioinformatics are either *(1)* standard routines that were adapted to the field or *(2)* algorithms particularly developed in the field. Examples of the former are most of the feature selection methods (see below), clustering and classification methods, and for the latter are CLICK, CAST and Gene-Shaving for clustering gene-expression data and BLAST for matching sequences. Algorithms, which belong to the first group are generic and perhaps not adequate to

handle genomic data and the others are mostly domain-specific that probably cannot be generalized to other data types.

It is customary to divide data mining methods based on the stages of the data analysis in which they are applied (i.e., preprocessing or categorization), and whether they are supervised or not. One can therefore classify data mining methodologies into four main classes: *(1)* unsupervised-preprocessing, *(2)* unsupervised-categorization (clustering), *(3)* supervised-preprocessing and *(4)* supervised-categorization (classification).

The focus of this research is to investigate and develop data mining techniques, which belong to the four partitions described above. In particular, much emphasis is put on unsupervised methods, and studying relatively less explored fields.

Unsupervised-Preprocessing

The first step of most data analysis procedures is usually preprocessing. It functions mostly to prepare the dataset (e.g., normalization, missing values imputation), eliminate noise, filter out irrelevant instances or features that describe each instance, and to reduce the dimensionality of the data.

One popular unsupervised approach to achieve normalization, dimensionality reduction and noise filtering, is *feature extraction.* However, ICA, PCA, SVD and other extraction methods transform **all** features to a lower dimension space, and do not allow attaching meaning to some relevant features in the set as in *feature selection*.

Although most analysts, often inattentively, do apply some unsupervised schemes (e.g., filtering out thousands of genes with small variance), surprisingly, only a few solutions have been suggested to select features in an unsupervised manner. Most of them are very naïve e.g., range, fold-change, threshold, entropy and variance calculated on each feature individually.

The importance of *unsupervised feature selection* to Bioinformatics, and the absence of unbiased, efficient, stable and effective solutions to address this issue, was our rationale to develop an *unsupervised feature filtering* algorithm (UFF). UFF differs from other unsupervised selection schemes in the following aspects *(1)* It does not involve a target function as the selection criterion and *(2)* it considers the interplay of all features. UFF scores each one of the features according to its contribution to the SVD-entropy of the dataset. Scoring a feature is based on a leave-one-out principle.

We have shown on several datasets (e.g., gene-expression, amino-acid composition counts) that: *(1)* Selection of only a few features according to UFF leads to improved clustering results as compared to other unsupervised methods or to using the complete set, *(2)* UFF is robust even in cases of severe information loss, *(3)* selected features are correlated with biological importance in cancer studies, *(4)* an a-priori criterion can provide an estimation of the effectiveness of the method for a given dataset and *(5)* the method can be generalized to select instances rather than features.

Unsupervised-Categorization (Clustering)

Clustering algorithms aim to find distinctive and, hopefully, relevant groups of instances in the dataset. This popular approach was very effective in clustering genes and tissues in gene-expression experiments and proteins according to their sequence similarities. Two aspects in clustering, addressed in our work, are the global considerations in clustering and their evaluation.

Global considerations in clustering: As noted above, some algorithms were initially developed to handle genome-specific data, while others are general machine learning procedures. One of the most popular standard routines is the agglomerative hierarchical algorithm, which is applied in a vast majority of cases. A clear limitation of this algorithm is its tendency to neglect global factors. In order to embed global considerations in clustering, we developed two algorithms: *(1) TDQC:* a novel Top-Down hierarchical algorithm based on genuine density of the data-points, and *(2)* a global-local ('glocal') variation of the agglomerative algorithm, which is based on all relationships within the data (all distances). A comprehensive analysis shows that the two new algorithms outperform other divisive and agglomerative methods. This assessment was tested in multiple domains, including gene-expression, stock trade records and functional protein families.

Clustering evaluation: High-throughput biological data is often noisy and of extremely large size (both in number of instances and in number of features). Therefore, manual or visual evaluation of clustering results is practically impossible. As the variability of data is so broad, no single clustering algorithm can always be effective, and preferred to others. Furthermore, because many algorithms encounter various limitations, determining the best solution is a very challenging task. We therefore designed and implemented three algorithmic and software frameworks that provide platforms to handle the above mentioned obstacles: *(1) The Clustering Algorithms Optimizer,* which is a completely unsupervised set of procedures that scan the clustering solutions space and identify the optimal solution vis-à-vis an internal measure, based on the Bayesian Information Criterion (BIC). This methodology performs well and overcomes intrinsic limitations of many clustering algorithms that rely on some predetermined parameters or involve nondeterministic factors. *(2) COMPACT: Comparative Package for Clustering Assessment-* A methodology and set of procedures that allow statistical and visual options to compare many algorithms and asses their results. (3) *ClusTree: A* graphical software package to analyze and compare hierarchical clustering.

Supervised-Preprocessing (feature selection)

As opposed to the unsupervised selection methods described above, *supervised feature selection* algorithms have been extensively studied and applied. Popular examples are: forward insertion, backward elimination, stepwise selection and ranking according to statistical criteria (e.g., *t*-test).

Supervised-Categorization (classification)

Classification algorithms learn patterns in the data, according to a training set, and try to induce a generalization rule, which fits the entire data. As in clustering, Support Vector Machine (SVM), decision trees, and other supervised methods, were effective in classifying gene-expression or sequence-based data.

In a research combining the two steps of supervised analysis (selection and classification), we hypothesized that high-level functional groups of proteins may be classified by a very small set of biochemical global features (e.g., molecular weight, hydrophobicity, amino-acid composition). To test this assumption, proteins represented by those global features, were classified using SVM. Furthermore, using various feature selection strategies, the contribution of specific subsets of features to the classification quality was thoroughly investigated. Our results show that a small set of global features that, sometimes, can be further reduced, provides effective information for protein family classification. Moreover, we found that a combination of global and local sequence features significantly improves classification performance.

Several general motivations led to these studies. First, when possible, unsupervised algorithms were preferred. The rationales to prefer the unsupervised approach in genomic data mining are: *(1)* Being less biased deriving from irrelevant factors, allowing for the emergence of more reliable and sometimes surprising results. *(2)* Only a relatively small portion of genomic items are fully labeled. *(3)* The train-test splitting, performed in supervised methods, is often problematic, (might cause over-fitting, sampling bias etc.).

The algorithms we suggest are based on mathematical and statistical principles, ignoring any specific biological considerations. Therefore, our methods are generic and not limited to a specific biological dataset, yet they are all well suited for large scale biological data. We should note however, that since biological understanding was our motivated force, inference was a principal focal point part of each study.

In addition, in data mining, and particularly in the cases of noisy biological data, it is very unlikely to expect a "one size fits all" practice, in other words, for every particular case a different algorithm and configuration should be preferred. Therefore, we put a strong emphasis on developing appropriate evaluation methods.

Most of the directions we explored have not been well studied in the literature, in particular, unsupervised feature filtering and global hierarchical algorithms. This research suggests that currently overlooked approaches should not be neglected. Surprisingly, these methods are shown to be effective when exploring genomic and proteomic data.

Our research was guided by realistic and applicative motivations, not limited only to theoretic perspectives. As a result, all our algorithms were applied to experimental datasets. In all cases, a software tool was developed for the corresponding algorithm. For instance, the COMPACT package, which has been made freely available for academic usage, has been accessed to date, more than 6,000 times, downloaded more than 800 times and was the basis for two graduate courses. Because potential users of these tools may be biologists or medical researchers who are not data mining experts, providing intuitive graphical and user-friendly applications was essential.

Finally, throughout the research, we were motivated to follow the principle of Occam's razor. Hence, we favored solutions that are easy to comprehend and fast to implement. Additionally, a main focus of our research was to find a minimal set of features or parameters that describe hidden patterns in the data.

Publications included in this thesis:

Chapter 2

[2A] Roy Varshavsky, Assaf Gottlieb, Michal Linial and David Horn. "*Novel Unsupervised Feature Filtering of Biological Data*" (2006, ISMB, Bioinformatics 2006, 22(14):e507-513).

[2B] Roy Varshavsky, Assaf Gottlieb, David Horn and Michal Linial. "*Unsupervised Feature Selection under Perturbations: Meeting the Challenges of Biological Data*" (2007, Bioinformatics, 23, 3343-3349).

Chapter 4

[4A] Roy Varshavsky, Michal Linial and David Horn. "*Clustering Algorithms Optimizer: A Framework for Large Datasets*" (2007, ISBRA, Lecture Notes in Bioinformatics (4463), 85-96).

[4B] Roy Varshavsky, Michal Linial, David Horn. "*COMPACT: A Comparative Package for Clustering Assessment*" (2005, ISPA, Lecture Notes in Computer Science (3759), 159-167).

Chapter 5

[5A] Roy Varshavsky, Menachem Fromer, Amit Man and Michal Linial. "*When Less is More: Improving Classification of Protein Families with a Minimal Set of Global Features*" (2007, WABI, Lecture Notes in Computer Science (4645), 12-24).

Other works (peer-reviewed posters and presentations)

Assaf Gottlieb*, **Roy Varshavsky***, Michal Linial and David Horn. “*Unsupervised Feature Filtering and Instance Selection*” (*equal contribution, see Appendix).

Roy Varshavsky, Yoel Bogoch, Marta Weinstock-Rosin, Michal Linial and David Horn. “*Two-step Clustering Algorithm: An Application to Gene Expression of Prenatal Stressed Rats*” (Israeli Bioinformatics Symposium, 2004)

Roy Varshavsky, David Horn and Michal Linial. "*Recursive Top-Down Quantum Clustering of Biological Data*" (2006, ISMB, PLoS Track, Oral presentation)

Roy Varshavsky, David Horn and Michal Linial. “*A Suite of Unsupervised Machine Learning Algorithms*” (ECCB, 2006, the 2nd Israeli Innovation Summit, 2007)

Content

Chapter 1

Introduction

Imagine a world in which doctors could diagnose most malicious diseases way before the apparent symptoms can be observed. In such a world, preventive medicine can really be efficient. Moreover, in cases when the illness has progressed, doctors would prescribe the right dose of the right drug at the right time, reducing the current trail-and-error practices. Such a personalized medicine is still a hope, yet science has made some significant steps toward fulfilling this ambitious goal.

The most significant progress in recent years was made in improving diagnosis. This was allowed by embedding genomic factors in the analysis. The study of the genome and the proteome has undergone some revolutionary advances due to the introduction of new technologies that can rapidly and accurately measure thousands of records. A prominent example of these technologies is the DNA-chip that simultaneously measures the expression of the entire genome (tens of thousands of genes), in a living-organism's tissue.

A typical experimental setting that uses these chips often consists of samples of dozens of tissues, taken from different individuals (either human or other species). A researcher that analyzes such an experiment may ask several questions: does the overall genomic signature correlate with some biological understanding (i.e., are there some meaningful patterns in the data?); according to the expression of the genes, can distinctive groups of instances be observed? Are there groups of genes that are similarly expressed? Is there a minimalist subset of the genes in the array that may be used for identifying a given biological phenomenon or a disease? The last question is of high importance, as it may lead to better understanding of the underlying processes involved in that phenomenon. Moreover, the genes included in such a set may serve as biomarkers for accurate diagnosis. Indeed, in recent years some diagnostic chips (e.g., for breast cancer) have been introduced.

Our research was motivated by these questions. In particular, our aim was to extract hidden patterns out of large-scale genomic and proteomic data and to suggest computational methods for revealing relevant groups (or subgroups) of entities in them.

1.1. Thesis outline

Chapter 1, the introduction, gives a brief description of the tools and the specific practices that were used throughout the research. Section 1.2 includes technical definitions; sections 1.3 and 1.4 describe the datasets and data types that were analyzed, respectively. Sections 1.5 and 1.6 include a high-level overview of data mining in general and data mining for genomic and proteomics in particular. The following sections include presentation of several data mining procedures that are applied in bioinformatics and are relevant to this research.

Chapter 2 describes a novel framework for *unsupervised feature filtering* (UFF). UFF is a unique approach for selection of features without previous knowledge of their classification, yet considering the interplay of all features. Selection according to this approach is effective and stable under incomplete information. It leads to interesting biological observations (Varshavsky, et al., 2006; Varshavsky, et al., 2007).

Chapter 3 includes an analysis of hierarchical unsupervised categorization (clustering). This analysis shows that global considerations, embedded in hierarchical clustering, can improve clustering results and reveal meaningful patterns in data. Furthermore, two new procedures, TDQC (Top-Down-Quantum-Clustering) and 'Glocal' (Global-Local) algorithms are suggested and shown to be highly effective for clustering data of different domains.

Chapter 4 includes a number of tools used for clustering evaluation. ”Clustering algorithms optimizer”, based on an internal criterion is suggested for usage in an unsupervised internal assessment (section 4.1). Two tools (COMPACT and ClusTree) are based on external criteria, and provide visual comparison and quantitative assessment routines (sections 4.2 and 4.3, respectively). These tools, providing access to several clustering algorithms (partitioning and hierarchical), were successfully applied to various datasets.

Chapter 5 presents a study based on supervised learning practices (feature selection and classification). This study shows that often only a small set of global features suffices to perform functional classification of proteins.

Chapter 6 concludes the thesis and provides a unifying discussion of our studies. A summary of conclusions common to the different studies is provided. Supplementary information completes the dissertation (for simplicity and coherence, relevant references are provided at the end of each chapter).

In order to orient the reader, an arranged view of the chapters, according to the different stages of the data mining process, is presented in Table 1.

Table 1: Methods applied in standard data mining application, arranged according to the stages in the analysis process (rows) and appearance in the thesis (columns)

Step	Analysis	Introduction	Ch. 2	Ch. 3	Ch. 4	Ch. 5
1	Data Preparation	1.5.1	+			+
2	Data Representation	1.4				
	Feature-space	1.4	+	+	+	+
	Similarity-space	1.4		+	+	+
3	Preprocessing	1.5.2, 1.7.1, 1.7.3				
	No			+		
	Extraction	1.7.1		+	+	
	Selection	1.7.1, 1.7.3	+			+
4	Categorization					
	Clustering	1.7.2				
	Hierarchical	1.7.2.1				
	No	1.7.2.1	+	+	+	
	Bottom Up	1.7.2.1		+	+	
	Top Down	1.7.2.1		+	+	
	Evaluation	1.7.2.2				
	Internal	1.7.2.2			+	
	External	1.7.2.2	+	+	+	+
	Classification	1.7.3				+
5	Biological Inference		+	+		+

1.2. *Notation, definitions and assumptions*

Data: Let us consider a dataset comprising n instances $\boldsymbol{A}_{[mXn]} = \{\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_i, \ldots, \bar{A}_n\}$, where each instance, or observation, $\bar{A}_i$ is a vector of m measurements or features describing it.

Categorization: Categorization is defined as systematically arranging instances into specific groups. In a categorization task (Figure 1), every instance, $\bar{A}_i$ has a label Y_i, where Y_i is a categorical parameter ($Y_i \in \{\chi_1, \chi_2, \ldots \chi_k\}$). A categorization algorithm is a function f that assigns a label, $\tilde{Y}_i$ to an instance ($f\ (\bar{A}_i) \rightarrow \tilde{Y}_i$). An algorithm is usually evaluated according to how well each predicted label, $\tilde{Y}_{i,}$ can be mapped to the true label Y_i.

Throughout this work we refer to any algorithm that assigns instances to labels as "a categorization algorithm". We distinguish between clustering (unsupervised categorization) and classification (supervised categorization) algorithms.

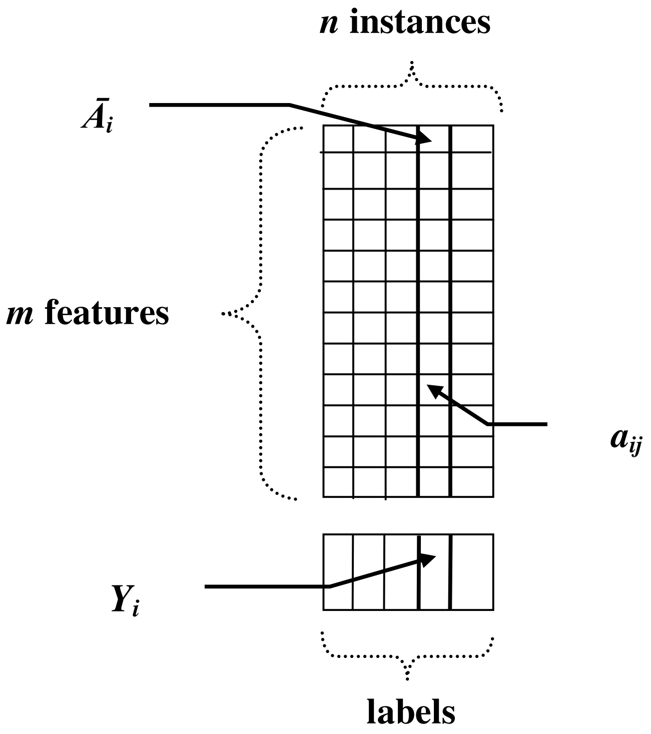


Figure 1: Schematic representation of a dataset in a categorization task, comprising *n* instances, each described by *m* measurements, and a label Y_i assigned to it.

Assumptions: We assume that $\forall a_{ij} \in \mathbb{R}$ (all records are real numbers), and that $\boldsymbol{A}$ is complete (i.e., there are no missing values). In data preparation and UFF descriptions (sections 1.5.1, and 2.2, respectively) we discuss cases with missing values.

Terminology: This work relates to *genomic* (i.e., belonging the genome) and *proteomic* (i.e., belonging to the proteome) analyses. The scientific field, in which genomic and proteomic problems are studied through computational and algorithmic tools, is called *Bioinformatics.*

1.3. High-throughput genomic and proteomics experiments

In the last several years, some high-throughput technologies that collect genomic and proteomic data were introduced. These technologies are considered as breakthroughs since they allow fast sequencing, recording the expressions of thousands of genes simultaneously, locating interactions between tens of thousands proteins and DNA and measuring many properties of the proteome. Most of these technologies were initially developed in universities and research centers, but have become readily available, cheaper and more reliable once produced by commercial companies. In conjunction with the data-collection feasibility, the progress in storage management (i.e., databases) and information transfer (in particular, the internet revolution), have facilitated the reposition and enabled more efficient retrieval of data. Therefore, numerous online genomic repositories have become publicly available.

We briefly describe in this section the high-throughput technologies of DNA and CGH chips. A more in-depth overview and descriptions of other methods (e.g., Haplotype Map, ChIP-on-chip, Mass Spectrometry, and protein chips) can be found in corresponding references. Data repositories that have been used in the research are also presented.

1.3.1. Technologies

1.3.1.1. Gene expression microarrays (DNA chips)

Gene expression microarray is probably the best known high-throughput technology that has been applied to genomic data. This technology allows for miniaturization of hybridization filters and as a result, measurement of thousands of different RNA molecules representing the expression of thousands of genes and even a complete genome.

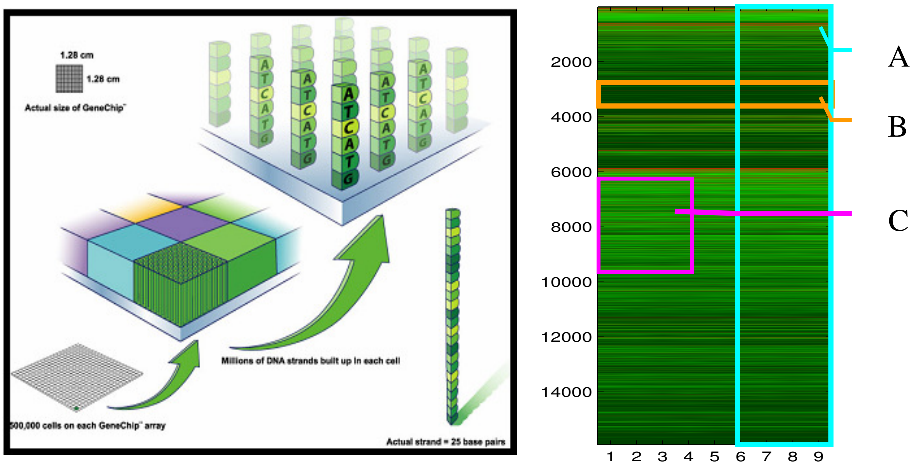

1.28 cm
1.28 cm
Actual size of GeneChip™
A T C A T G
Millions of DNA strands built up in each cell
500,000 cells on each GeneChip™ array
Actual strand = 25 base pairs
2000
4000
6000
8000
10000
12000
14000
1 2 3 4 5 6 7 8 9
A
B
C

Figure 2: The microarray concept (left) and results (right). Shown are (A) samples, (B) genes, and (C) two-way comparisons. Data taken from an experiment done in the lab (Affymetrix RAE 230A, chip)

Researchers can either use in-house, per-demand microarrays or standard, off-the-shelf chips that are produced by one of the commercial manufacturers (e.g., Affymetrix, Agilent). The introduction of commercial DNA-chips has significantly reduced the cost of each experiment and its complexity, and accelerated the popularity of gene-expression studies. In addition, recent studies showed that this technology has reached maturity (Shi, et al., 2006), sufficing to yield reliable results (Klebanov and Yakovlev, 2007).

A typical experimental configuration would consist of several chips, allowing a multipurpose analysis (Figure 2): (A) disease diagnostics based on samples comparison (between different tissues or different conditions). (B) Gene discovery and taxonomy based on genes comparison (genes that similarly vary along the different experimental settings) and (C) Two-way comparisons (finding groups of genes whose expressions correspond to subsets of the samples provided).

As gene-expression experiments have become so popular and the potential of analysis is so wide, most of the algorithms and tools presented in this work were applied to this type of data.

1.3.1.2. Comparative Genomic Hybridization (CGH)

Similarly to DNA chips, the relatively new CGH technology provides simultaneous recording of thousands of genomic changes. However, while DNA chips measure affinity for cell molecules and the chip's probes, the CGH classifies copy number alternations as DNA gains and losses, and its results are considered to be more reliable than DNA chips.

Despite the fact that this technology has not reached maturity yet, it is becoming very popular, with increasing number of experiments utilizing it (Baudis and Cleary, 2001).

We have applied our novel filtering method (UFF) to a noisy, incomplete CGH benchmark (Varshavsky, et al., 2007, section 2.2), and showed that the method can successfully handle this data, suggesting some potentially interesting biological interpretations.

1.3.2. Repositories

In parallel, and as a result of the availability in high throughput technology, several publicly available data collections hold thousands of experimental results. These open repositories include results of gene-expression experiments, many sequenced genomes and functional, cellular and other annotations related to genes and proteins.

1.3.2.1. Gene expression

Microarray experimental results can be easily accessed in several gene expression repositories. Collections such as Stanford Microarray database (Ball, et al., 2005), Gene Expression Omnibus of NCBI (Barrett, et al., 2007) and ArrayExpress at EBI (Sarkans, et al., 2004), archive thousands of datasets (in October 2007, the numbers of datasets were 15238, 12376 and 2644, each holding dozens of cases). Each set includes raw and processed data, corresponding publication and supplementary information. In addition, open repositories holding the fast growing CGH data have become available (e.g., the progenetix database with 16252 cases from 634 publications, Baudis and Cleary, 2001).

1.3.2.2. Sequences

Today it has become easier to gain access to sequenced genomes. Sequence related databases provide different perspectives on sequenced data. While in 2003, about 1 million sequences were stored in the UniProt database (release 1.0), today (release 12) it contains about five million protein sequences, and this collection is expected to grow (Wu, et al., 2006). This database also provides non-redundant subsets of the entire list (UniRef90 and UniRef50) yielding a reduction to 3 and to 1.5 million sequences, respectively. In those instances, no two sequences are permitted to share more than 90% or 50% sequence identity, respectively.

The Pfam database holds a collection of multiple sequence alignments and protein domains, classified into around 9000 families (Finn, et al., 2006). Other databases hold a clustered view of the genome (Kaplan, et al., 2005) and structural information (Berman, et al., 2000; Balaji, et al., 2001; Bhat, et al., 2001).

1.3.2.3. Annotations

In addition to the sequences of each protein or gene, several databases keep functional, cellular compartment and other annotation of genes (e.g., GO, Camon, et al., 2004) or proteins (e.g., UniProtKB, Kriventseva, et al., 2001). Several of these annotations are manually curated, while other are based on a combination of biological understanding and algorithmic power. This information is usually considered as an ‘expert’ view of the instances, and thus is utilized for evaluation of categorization algorithms.

Categorization of sequenced proteins is presented and assessed in chapters 3 and 5 (Varshavsky, et al., 2007). A discussion of the limitation of this ‘expert’ based evaluation and the capability of our approach to reduce it is provided in the conclusions of these chapters.

1.4. Data types

Data may come in two possible representations: feature-space or similarity-space.

A [*mXn*] *feature-space* matrix represents each instance according to its features or attributes. For example: Gene expression (Figure 2), 3D coordinates of protein structures, global features (hydrophobicity, length, Cai, et al., 2003; Syed and Yona, 2003; Varshavsky, et al., 2007).

A [*nXn*] *similarity* (or dissimilarity) matrix represents each instance by its similarity (dissimilarity) to another instance (Figure 3). When a distance function defines the dissimilarity between instances, this square representation leads to a symmetric matrix. The popular distance functions are:

Norm *l*1 distance $\|d(x,y)\|_1 = |x_1 - y_1| + \ldots + |x_m - y_m|$

Norm *l*2 (Euclidian) distance $\|d(x,y)\|_2 = \sqrt{|x_1 - y_1|^2 + \ldots + |x_m - y_m|^2}$

Correlation coefficient distance $\|d(x,y)\|_{ccd} = 1 - \frac{Cov(x,y)}{\sigma_x \sigma_y}$

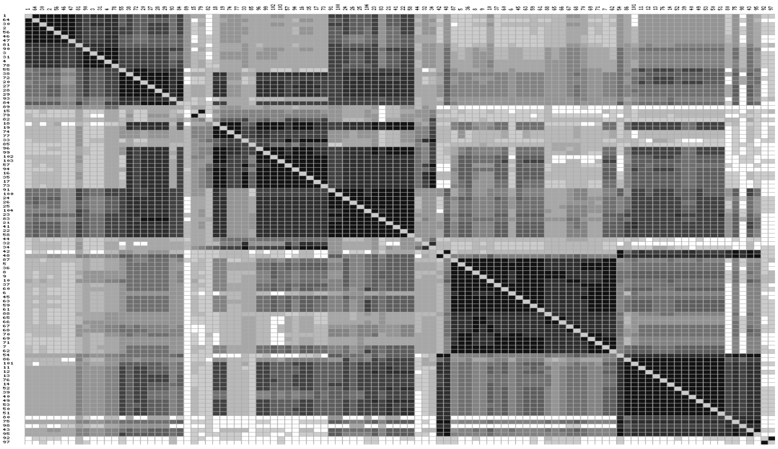


Figure 3: Distance (translated to E-score) between protein sequences, as displayed by the ProtoMesh web-tool (www.protonet.cs.huji.ac.il/mesh2)

In some cases, similarity between elements are directly extracted (without transforming from feature-space). Examples are: BLAST (Altschul, et al., 1997) or Smith-Waterman (Smith and Waterman, 1981; Smith, et al., 1981) matrices in proteomics. Given only the similarity, feature-space cannot be reconstructed (except for approximations as in Multidimensional Scaling, Kruskal and Wish, 1981). In the context of categorization, most algorithms operate on distances between elements (e.g., agglomerative hierarchical algorithm), while others on feature-space (K-Means, QC, see section 1.7.2, chapter 3 below and Duda, et al., 2000).

1.5. Data Mining

Knowledge Discovery in Databases (KDD), known as "Data Mining", is an approach based on statistical, algorithmical and other mathematical methods used to extract nontrivial information from data (Frawley, et al., 1992). In recent years, large-scale data has become available in many scientific and applicative domains. Due to the complexity of the data and the questions, there is a strong emphasis on applying automated routines with the least amount of user interaction. Therefore, data mining is a flourishing field. Examples of data applications are: marketing (CRM, customers segmentation, markets trends, customers loyalty), stocks (trades patterns, associates stocks or correlations with external factors), text mining (document classification), intelligence, internet (web search, ad-sense), finance (fraud detection) and of course in biology (Azuaje, 2006).

Analysis based on data mining includes several stages, that can vary according to the data at hand and the task. The procedures listed below are frequently used in data mining of biological datasets.

1.5.1. Data preparation

Experimental data are often noisy, not fully reliable and incomplete. In order to improve the accuracy of the analysis, several data preparation procedures are usually applied. The standard methods are values transformation (e.g., log-transform in DNA chips), handling categorical features, discretization and normalization (e.g., to mean 0 and standard deviation 1). Other quality control methods are background elimination (e.g., dummy probes in DNA chips that do not attach to any molecule, whose recorded values should be deducted from the values of the real probes) and replicates based analysis (Singh and Nagaraj, 2006). In our work we assume that data's origin is one meta-distribution, so that genuine values carry significant information. Therefore, we do not address in this work data transformation and it is not specifically applied nor studied. Moreover, assuming all features are real, handling categorical features is not required. Implicitly we assume that the data have already passed the quality control procedures.

Absent values becomes a challenge when handling experimental data, particularly in experimental high-throughput genomic datasets (de Brevern, et al., 2004; Scheel, et al., 2005). Several methods were developed to address this issue (Troyanskaya, et al., 2001; Gan, et al., 2006; Tuikkala, et al., 2006; Hua and Lai, 2007), starting from the naive zero or average replacements (i.e., each missing value is replaced by 0 or by the average of all present values in the set, respectively), to more sophisticated ones (e.g., KNNImpute, where each missing value is replaced by the standard average of samples of the K nearest neighbors of a relevant feature, Troyanskaya, et al., 2001). This issue is discussed in chapter 2.1 (Varshavsky, et al., 2007), where the UFF method is evaluated in noisy datasets. Applying it to incomplete data that has undergone several imputation procedures, it is shown that the method can sustain high accuracy levels even after substantial loss. Furthermore, it can assess in an unsupervised manner the various imputation procedures.

1.5.2. Preprocessing

Data preparation is often followed by data preprocessing. Various procedures may be included at this stage, but most of them have similar aims: dimensionality reduction, outliers removal, selection of the most informative features and noise filtering (also known as increasing signal to noise ratio).

All these methods aim to reduce the size of data analyzed and minimize the complexity of the problem. Compression of datasets is essential to incorporate algorithms that cannot perform well on large-scale data, or have generalization limitations in higher dimensions. In addition, an accurate partition between relevant and obscuring elements can improve interpretation extracted from the analysis. Descriptions of several representative preprocessing methods, relevant to the research, appear in sections 1.7.1 and 1.7.3.

1.5.3. Model fitting

After preprocessing of data, several analysis procedures can be applied. These methods can be classified according to several criteria: unsupervised (do not rely on labels), or supervised (using a labeled train set), categorization (where the labels are categorical variables) or regression (labels are continuous variables).

The focus of this research is unsupervised and supervised categorization (clustering and classification, respectively). Regression, association rules, generative models, decision trees and other popular data mining techniques lie outside the scope of this work.

1.6. Data mining in genomic and proteomics

Breakthroughs in technology and the improved effectiveness of data mining lead to what has been described as *"a paradigm shift in biological investigation, such that the bottleneck in research is shifting from data generation to data analysis*" (Sherlock, 2000). The following paragraphs provide a brief review of the special characterizations of genomic and proteomic data.

The accelerated growth in the size of the UniProt repository from one to five million sequences in less than five years (Wu, et al., 2006), which is typical of biological data, poses a significant challenge which is almost unsolvable by traditional research techniques. For example, the 5-fold multiplication in the size of the repository leads to 25-fold more calculations in clustering methods that involve computation of all relations between elements (e.g., Kaplan, et al., 2005). Data observation, analysis and inference need therefore advanced procedures.

In many fields where data mining is applied, the number of features is quite limited. For example, in marketing applications, records collected from potential customers may amount to only a few hundreds. However, the number of features in genomic data is tremendous (e.g., tens of thousands gene-expression records per tissue). Simplistic observation of such data is impossible. In addition to that, in tasks where learning about the instances is required, the number of dimensions (genes) is significantly bigger than the number of instances (samples). Such a phenomenon is referred to as the "curse of dimensionality" problem (Bellman, 1961), in which in learning rules, an exponential increase in the number of instances is required when adding extra dimensions. The major challenge of this problem is that no experimental setting can be learnt and generalized unless thousands of instances are measured (Ein-Dor, et al., 2006). Dimensionality reduction is therefore an essential preprocessing procedure.

A fundamental characteristic of biological data is the great number of intervening factors. These factors include the underlying variance between biological observations, differences between experimental settings, technology (that although becoming more stable, still has various flaws (e.g., Irizarry, et al., 2003), imprecision of scanning devices, recording and software. All of them increase the relative noise flux in data, and therefore call for efficient noise filtering techniques.

Another obstacle, almost exclusive to this field, is the difficulty of inference. While in other disciplines (e.g., in document mining), it is relatively easy to assign experts to validate the results and provide a ‘ground-truth’ benchmark, in genomic and proteomics, current knowledge is still inadequate; hence, many proteins are still unlabeled.

1.6.1. Applications

Categorization is the most common data mining practice applied in bioinformatics. Popular categorization tasks are grouping instances (samples) according to their gene-expression pattern (Golub, et al., 1999; Sharan, et al., 2002; D'Haeseleer, 2005), grouping genes that are similarly expressed along different experimental settings (Spellman, et al., 1998), grouping proteins according to their sequence (Kaplan, et al., 2004; Kaplan, et al., 2005) or other properties (Cai, et al., 2003; Varshavsky, et al., 2007). These tasks can be applied either in unsupervised or supervised manner.

Many dimensionality reduction, noise filtering and feature selection methods have been suggested to address challenges presented by genomic and proteomic data. A major part of this research was devoted to study, analyze and develop efficient, data-driven compression methods.

Other efficient data mining procedures, not discussed here, are sequence motifs search (Skoufos, 1999; Kriventseva, et al., 2001; Kunik, et al., 2005), systems and network dependencies analysis (known as interactome study, Fattore and Arrigo, 2005; Singh and Nagaraj, 2006) and text/literature mining. During recent years much effort is devoted to dig into the abundance of data covered in the literature (Hirschman, et al., 2002; Jensen, et al., 2006). These efforts, based on knowledge from NLP (Natural Language Processing), archiving and document classification, try to extract interesting biological knowledge.

1.7. Algorithms

Data mining algorithms in bioinformatics are either: *(1)* routines developed in other domains (e.g., Physics, Mathematics, Statistics and Computational Neuroscience), that were adapted to the deal with genomic or proteomic problems, or *(2)* algorithms that were specifically designed to mine gene-expression or sequence based data. Examples of routines developed in other domains are most of the feature selection methods (Saeys, et al., 2007), clustering and classification methods (D'Haeseleer, 2005). Examples of designated methods are CLICK (Sharan and Shamir, 2000), CAST (Ben-Dor, et al., 1999; Ben-Dor, et al., 2001) and Gene-Shaving (Hastie, et al., 2000) for clustering gene-expression data, and BLAST (Altschul, et al., 1997) for matching sequences. Algorithms which belong to the first group are generic and perhaps not adequate to handle genomic data and the others are mostly domain-specific and probably cannot be generalized to other data types.

It is customary to divide data mining methods according to stages of the data analysis in which they operate: preprocessing or categorization, supervised or not. One can therefore classify data mining methodologies into four main classes: *(1)* unsupervised-preprocessing, *(2)* unsupervised-categorization (clustering), *(3)* supervised-preprocessing and *(4)* supervised-categorization (classification).

1.7.1. Unsupervised preprocessing

Preprocessing is applied to facilitate analysis of the data. Preprocessing methods are either applied for preparation (see section 1.5.1, above), or for dimensionality reduction. The significantly large size of the data gathered using high-throughput technology makes dimensionality reduction a necessity. According to (Guyon and Elisseeff, 2003; Saeys, et al., 2007), the major objectives of dimensionality reduction are: reducing over-fitting, improving model performance, lowering runtime and other costs and providing a better insight of underlying processes. Dimensionality reduction methods are described as *feature extraction* or *feature selection*. Feature extraction methods transform **all** features to a lower dimension space and feature selection methods select some relevant features in the set.

In our research, Singular Value Decomposition (SVD) was used for *extraction*. SVD represents any real matrix $\boldsymbol{A}$ as a product of three matrices $\boldsymbol{A}=U\Sigma V^T$, where U and V are orthonormal matrices and Σ is a diagonal matrix whose eigenvalues s_i (singular values) appear in decreasing order (Figure 4). The columns of *U* and *V* define two independent vector spaces. This decomposition is unique (up to overall phases) and holds for any real matrix of size *m* by *n*. The number of non-zero entries in *Σ* equals the rank of ***A***. A common practice of application of SVD for dimensionality reduction is replacing *Σ* with a truncated version *Σ',* where only a small number *r,* of leading singular values, is retained and the rest are replaced by zeros. The resulting reconstructed matrix $\boldsymbol{A}'=U\Sigma'V^T$, is the best least-mean-squares approximation of ***A*** obtainable by any matrix of rank *r*.

An alternative utilization of the SVD procedure consists of focusing attention on the matrices *U* and *V* which, in gene-expression datasets, form gene and sample spaces, respectively. It is within these spaces, now reduced to rank *r* that one can look for data patterns (Alter, et al., 2000; Alter and Golub, 2006; Horn and Axel, 2003; Wall, et al., 2003). Such an extraction application is presented in chapter 4 (Varshavsky, et al., 2005; Varshavsky, et al., 2007).

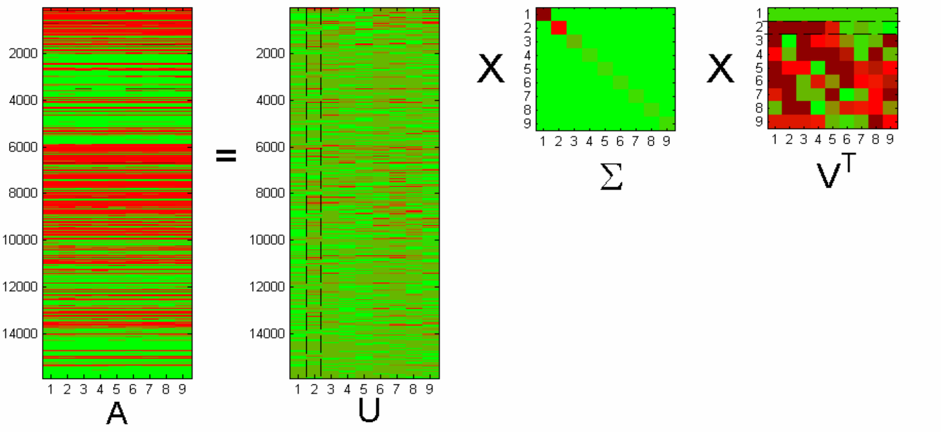


Figure 4: Visualization of SVD routine as applied to gene-expression data (see appendix 2)

There exist only a handful of unsupervised feature *selection* algorithms (Guyon and Elisseeff, 2003; Saeys, et al., 2007). As described in (Dy and Brodley, 2004), such methods can be applied at three different stages: before, during and after the clustering process. Methods which operate before clustering are referred to as *filter* methods. These methods are the least biased of all, as they do not depend on the clustering implementation. Common methods of unsupervised feature filtering rank features according to (1) their projection on the first principal component (Hartmann, 2006; Zou, et al., 2006) , (2) their normalized range,(3) entropy or (4) variance of the feature as calculated from its values on all instances (Guyon and Elisseeff, 2003; Herrero, et al., 2003).

We present a novel *unsupervised feature filtering* (UFF) framework, which differs from other unsupervised selection schemes in the following aspects: *(1)* it does not involve a target function as the selection criterion and *(2)* it considers the interplay of all features (chapter 2, Varshavsky, et al., 2006; Varshavsky, et al., 2007).

1.7.2. Clustering

In the last few years several clustering algorithms were found useful in handling genomic and proteomics data, for example: diagnosis of different conditions (between sick and healthy tissues), and classification to subtypes of a disease (Golub, et al., 1999; D'Haeseleer, 2005). An additional conclusion to the application of such algorithms to gene-expression data was the discovery of functional classes of genes among the thousands used in experimental settings (Eisen, et al., 1998). Furthermore, it became possible and useful to isolate groups of relevant genes that mostly contribute to a particular condition, a procedure called two-way or bi-clustering (Cheng and Church, 2000).

1.7.2.1. Clustering algorithms

Clustering algorithms are often classified as nonhierarchical (partitioning) or hierarchical. Nonhierarchical clustering algorithms define a complete partition of the data (for comprehensive reviews see Jain and Dubes, 1988; Duda, et al., 2000; D'Haeseleer, 2005). Because they suggest multiple levels of organization, hierarchical algorithms are perhaps the most popular clustering methods used (Spellman, et al., 1998).

Hierarchical methods can be further divided into Bottom-Up (BU, agglomerative) and Top-Down (TD, divisive) types (Jain and Dubes, 1988; Duda, et al., 2000; Planet, et al., 2001). BU algorithms start with every instance as a cluster and repeatedly merge clusters until a unified cluster is formed. TD methods work in the opposite direction and are rarely used for biological data. Algorithms can be alternatively classified by the following criteria *(1)* being deterministic or not, *(2)* being model-based or heuristic. Deterministic algorithms assume that the data was generated from a specific 'meta' distribution and the algorithms' objective it to reconstruct that distribution.

In this study we explored clustering algorithms in depth, focusing on hierarchical algorithms (chapter 3). In addition to that, we present two novel algorithms: *(1)* TDQC: Top-Down-

Quantum-Clustering algorithm, derived from QC (Quantum-Clustering) algorithm that was successfully applied to gene-expression data (Horn and Axel, 2003) and *(2)* a global-local ('glocal') variation of the agglomerative algorithm which is based on all relationships within the data (all distances).

1.7.2.2. Evaluation

Since different results can be obtained by different clustering algorithms, evaluation of this variety is an essential step of the analysis (Handl, et al., 2005; Varshavsky, et al., 2005). Other factors influencing evaluation and inference are: *(1)* the number of clusters contained in the dataset. Clustering algorithms usually require selecting a set of parameters, turning each application into a set of subjective choices. If no prior knowledge is available, assessing the correct number of clusters (e.g., as required by the K-Means algorithm), is almost impossible. Other algorithms do not explicitly accept the number of clusters as an input; however this number is directly derived from their parameters. *(2)* Algorithms such as K-Means, and others, being nondeterministic, are inconsistent as they depend on starting points and other stochastic factors.

Clustering assessment can be based on internal or external measurements. Internal criteria evaluate results solely on the data distribution and clustering partitions. In chapter 4.1 (Varshavsky, et al., 2007) we adopt the Bayesian Information Criterion (BIC), a model-based driven internal criterion (Fraley and Raftery, 1998) to compare between different algorithms, and select the optimal solution.

External criteria evaluate clustering results according to the labels of the instances, as assigned by experts. This post-analysis evaluation reflects the algorithm – real-world correspondence. Evaluations based on external criteria are presented and discussed in chapters 4.2 (Varshavsky, et al., 2005) and 4.3.

1.7.3. Supervised learning: Feature selection & Classification

In supervised learning, selecting the most relevant features and classifying the instances according to them, are two common procedures. Supervised selection approaches prioritize features according to the goodness of their fit to a classification task, and are usually defined according to logical relation to this task (Guyon and Elisseeff, 2003; Saeys, et al., 2007).

Selection methods are either *filter, wrapper* or *embedded.* Filter methods score each feature according to some criteria (e.g., *t*-test), and select the highest-scoring features. Wrappers try to optimize the classification task in an iterative way by adding a feature (forward insertion), removing a feature (backward elimination), adding or removing features (stepwise) or applying some more sophisticated, often randomized routines. Embedded methods are more related to the classification algorithm, selecting features that incorporate intrinsic consideration (e.g., selecting features with high correlation to the weights of the vectors resulted by SVM).

In bioinformatics, supervised learning is a very common strategy, and many of its aspects have been studied. In particular, genes have been ranked and selected according to how they classify instances to different cancer types (Khan, et al., 2001; Beer, et al., 2002). In our study, we employed various feature selection methods to a proteins dataset. In this dataset, proteins are characterized according to a few global features, derived from their sequence. By following a parsimony theme (central to this research), we showed that a very small set of features suffices to classify proteins to functional groups (chapter 5 and Varshavsky, et al., 2007).

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.

Alter, O. and Golub, G.H. (2006) Singular value decomposition of genome-scale mRNA lengths distribution reveals asymmetry in RNA gel electrophoresis band broadening, Proc Natl Acad Sci U S A, 103, 11828-11833.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res, 25, 3389-3402.

Azuaje, F. (2006) Review of "Data Mining: Practical Machine Learning Tools and Techniques" by Witten and Frank, BioMedical Engineering OnLine, 5, 51.

Balaji, S., Sujatha, S., Kumar, S.S. and Srinivasan, N. (2001) PALI-a database of Phylogeny and ALIgnment of homologous protein structures, Nucleic Acids Res, 29, 61-65.

Ball, C.A., Awad, I.A.B., Demeter, J., Gollub, J., Hebert, J.M., Hernandez-Boussard, T., Jin, H., Matese, J.C., Nitzberg, M., Wymore, F., Zachariah, Z.K., Brown, P.O. and Sherlock, G. (2005) The Stanford Microarray Database accommodates additional microarray platforms and data formats, Nucl. Acids Res., 33, D580-582.

Barrett, T., Troup, D.B., Wilhite, S.E., Ledoux, P., Rudnev, D., Evangelista, C., Kim, I.F., Soboleva, A., Tomashevsky, M. and Edgar, R. (2007) NCBI GEO: mining tens of millions of expression profiles--database and tools update, Nucl. Acids Res., 35, D760-765.

Baudis, M. and Cleary, M.L. (2001) Progenetix.net: an online repository for molecular cytogenetic aberration data, Bioinformatics, 17, 1228-1229.

Beer, D.G., Kardia, S.L.R., Huang, C.-C., Giordano, T.J., Levin, A.M., Misek, D.E., Lin, L., Chen, G., Gharib, T.G., Thomas, D.G., Lizyness, M.L., Kuick, R., Hayasaka, S., Taylor, J.M.G., Iannettoni, M.D., Orringer, M.B. and Hanash, S. (2002) Gene-expression profiles predict survival of patients with lung adenocarcinoma, Nat Med, 8, 816-824.

Bellman, R. (1961) Adaptive control processes - A guided tour. Princeton University Press, Princeton.

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

Ben-Dor, A., Shamir, R. and Yakhini, Z. (1999) Clustering Gene Expression Patterns, Journal of Computational Biology, 6, 281-297.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. (2000) The Protein Data Bank, Nucleic Acids Res, 28, 235-242.

Bhat, T.N., Bourne, P., Feng, Z., Gilliland, G., Jain, S., Ravichandran, V., Schneider, B., Schneider, K., Thanki, N., Weissig, H., Westbrook, J. and Berman, H.M. (2001) The PDB data uniformity project, Nucleic Acids Res, 29, 214-218.

Cai, C.Z., Han, L.Y., Ji, Z.L., Chen, X. and Chen, Y.Z. (2003) SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence, Nucleic Acids Res, 31, 3692-3697.

Camon, E., Barrell, D., Lee, V., Dimmer, E. and Apweiler, R. (2004) The Gene Ontology Annotation (GOA) Database--an integrated resource of GO annotations to the UniProt Knowledgebase, In Silico Biol, 4, 5-6.

Cheng, Y. and Church, G.M. (2000) Biclustering of Expression Data Intelligent Systems for Molecular Biology (ISMB). AAAI Press, 93-103

D'Haeseleer, P. (2005) How does gene expression clustering work?, Nat Biotechnol, 23, 1499-1501.

de Brevern, A., Hazout, S. and Malpertuy, A. (2004) Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering, BMC Bioinformatics, 5, 114.

Duda, R.O., Hart, P.E. and Stork, D.G. (2000) Pattern Classification. Wiley-Interscience.

Dy, J.G. and Brodley, C.E. (2004) Feature Selection for Unsupervised Learning, J. Mach. Learn. Res., 5, 845-889.

Ein-Dor, L., Zuk, O. and Domany, E. (2006) Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer, PNAS, 103, 5923-5928.

Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns, PNAS, 95, 14863-14868.

Fattore, M. and Arrigo, P. (2005) Knowledge Discovery and System Biology in Molecular Medicine: An Application on Neurodegenerative Diseases, In Silico Biology 5, 199 - 208.

Finn, R.D., Mistry, J., Schuster-Bockler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S.R., Sonnhammer, E.L.L. and Bateman, A. (2006) Pfam: clans, web tools and services, Nucl. Acids Res., 34, D247-251.

Fraley, C. and Raftery, A.E. (1998) How many clusters? Which clustering method? - Answers via Model-Based Cluster Analysis. Computer Journal. 578-588.

Frawley, W.J., Piatetsky-Shapiro, G. and Matheus, C.J. (1992) Knowledge discovery in databases - an overview, Ai Magazine, 13, 57--70.

Gan, X., Liew, A.W.-C. and Yan, H. (2006) Microarray missing data imputation based on a set theoretic framework and biological knowledge, Nucl. Acids Res., 34, 1608-1619.

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.

Handl, J., Knowles, J. and Kell, D.B. (2005) Computational cluster validation in post-genomic data analysis, Bioinformatics, 21, 3201-3212.

Hartmann, W. (2006) Dimension Reduction vs. Variable Selection. In, Applied Parallel Computing. 931-938.

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, research0003.0001 - research0003.0021.

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

Hirschman, L., Park, J.C., Tsujii, J., Wong, L. and Wu, C.H. (2002) Accomplishments and challenges in literature data mining for biology, Bioinformatics, 18, 1553-1561.

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

Hua, D. and Lai, Y. (2007) An ensemble approach to microarray data-based gene prioritization after missing value imputation, Bioinformatics, 23, 747-754.

Irizarry, R.A., Bolstad, B.M., Collin, F., Cope, L.M., Hobbs, B. and Speed, T.P. (2003) Summaries of Affymetrix GeneChip probe level data, Nucl. Acids Res., 31, e15-.

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

Jensen, L.J., Saric, J. and Bork, P. (2006) Literature mining for the biologist: from information retrieval to biological discovery, Nat Rev Genet, 7, 119-129.

Kaplan, N., Friedlich, M., Fromer, M. and Linial, M. (2004) A functional hierarchical organization of the protein sequence space, BMC Bioinformatics, 5, 196.

Kaplan, N., Sasson, O., Inbar, U., Friedlich, M., Fromer, M., Fleischer, H., Portugaly, E., Linial, N. and Linial, M. (2005) ProtoNet 4.0: a hierarchical classification of one million protein sequences, Nucleic Acids Res, 33, D216-218.

Khan, J., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C. and Meltzer, P.S. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nat Med, 7, 673-679.

Klebanov, L. and Yakovlev, A. (2007) How high is the level of technical noise in microarray data?, Biology Direct, 2, 9.

Kriventseva, E.V., Biswas, M. and Apweiler, R. (2001) Clustering and analysis of protein families, Curr Opin Struct Biol, 11, 334-339.

Kruskal, J.B. and Wish, M. (1981) Multidimensional scaling. Sage Publications, Beverly Hills.

Kunik, V., Solan, Z., Edelman, S., Ruppin, E. and Horn, D. (2005) Motif Extraction and Protein Classification 2005 IEEE Computational Systems Bioinformatics Conference (CSB'05). 80-85.

Planet, P.J., DeSalle, R., Siddall, M., Bael, T., Sarkar, I.N. and Stanley, S.E. (2001) Systematic Analysis of DNA Microarray Data: Ordering and Interpreting Patterns of Gene Expression, Genome Res., 11, 1149-1155.

Saeys, Y., Inza, I. and Larranaga, P. (2007) A review of feature selection techniques in bioinformatics, Bioinformatics, 23, 2507-2517.

Sarkans, U., Parkinson, H., Lara, G.G., Oezcimen, A., Sharma, A., Abeygunawardena, N., Contrino, S., Holloway, E., Rocca-Serra, P., Mukherjee, G., Shojatalab, M., Kapushesky, M., Sansone, S., Farne, A., Rayner, T. and Brazma, A. (2004) The ArrayExpress gene expression database: a software engineering and implementation perspective, Bioinformatics, bti157.

Scheel, I., Aldrin, M., Glad, I.K., Sorum, R., Lyng, H. and Frigessi, A. (2005) The influence of missing value imputation on detection of differentially expressed genes from microarray data, Bioinformatics, 21, 4272-4279.

Sharan, R., Elkon, R. and Shamir, R. (2002) Cluster analysis and its applications to gene expression data, Ernst Schering Res Found Workshop, 83-108.

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

Sherlock, G. (2000) Analysis of large-scale gene expression data, Curr Opin Immunol, 12, 201-205.

Shi, L., Reid, L.H., Jones, W.D., Shippy, R., Warrington, J.A., Baker, S.C., Collins, P.J., de Longueville, F., Kawasaki, E.S., Lee, K.Y., Lou, Y., Sun, Y.A., Willey, J.C., Setterquist, R.A., Fischer, G.M., Tong, W., Dragan, Y.P., Dix, D.J., Frueh, F.W., Goodsaid, F.M., Herman, D., Jensen, R.V., Johnson, C.D., Lobenhofer, E.K., Puri, R.K., Sherf, U., Thierry-Mieg, J., Wang, C., Wilson, M. and Wolber, P.K. (2006) The microarray quality control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements, Nat Biotechnol, 24, 1151 - 1161.

Singh, O.V. and Nagaraj, N.S. (2006) Transcriptomics, proteomics and interactomics: unique approaches to track the insights of bioremediation, Brief Funct Genomic Proteomic, 4, 355-362.

Skoufos, E. (1999) Conserved sequence motifs of olfactory receptor-like proteins may participate in upstream and downstream signal transduction., Receptors Channels, 6, 401-413.

Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences, J Mol Biol, 147, 195-197.

Smith, T.F., Waterman, M.S. and Fitch, W.M. (1981) Comparative biosequence metrics, J Mol Evol, 18, 38-46.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. (1998) Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization, Mol. Biol. Cell, 9, 3273-3297.

Syed , U. and Yona, G. (2003) Using a mixture of probabilistic decision trees for direct prediction of protein function., Proceedings of RECOMB . , 224-234. .

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman, R.B. (2001) Missing value estimation methods for DNA microarrays, Bioinformatics, 17, 520-525.

Tuikkala, J., Elo, L., Nevalainen, O.S. and Aittokallio, T. (2006) Improving missing value estimation in microarray data with gene ontology, Bioinformatics, 22, 566-572.

Varshavsky, R., Fromer, M., Man, A. and Linial, M. (2007) When Less Is More: Improving Classification of Protein Families with a Minimal Set of Global Features. In, Algorithms in Bioinformatics. 12-24.

Varshavsky, R., Gottlieb, A., Horn, D. and Linial, M. (2007) Unsupervised feature selection under perturbations: meeting the challenges of biological data, Bioinformatics, 23, 3343-3349.

Varshavsky, R., Gottlieb, A., Linial, M. and Horn, D. (2006) Novel Unsupervised Feature Filtering of Biological Data, Bioinformatics, 22, e507-513.

Varshavsky, R., Horn, D. and Linial, M. (2007) Clustering Algorithms Optimizer: A Framework for Large Datasets. In, Bioinformatics Research and Applications. 85-96.

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

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.

Wu, C.H., Apweiler, R., Bairoch, A., Natale, D.A., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Mazumder, R., O'Donovan, C., Redaschi, N. and Suzek, B. (2006) The Universal Protein Resource (UniProt): an expanding universe of protein information, Nucleic Acids Res, 34, D187-191.

Zou, H., Hastie, T. and Tibshirani, R. (2006) Sparse Principal Component Analysis, Journal of Computational & Graphical Statistics, 15, 265-286.

Chapter 2

Unsupervised Feature Filtering (UFF)

This chapter contains the following research papers:

[2A] **Roy Varshavsky**, Assaf Gottlieb, Michal Linial and David Horn. "*Novel Unsupervised Feature Filtering of Biological Data*" (2006, ISMB, Bioinformatics 2006, 22(14):e507-513).

[2B] **Roy Varshavsky**, Assaf Gottlieb, David Horn and Michal Linial. "*Unsupervised Feature Selection under Perturbations: Meeting the Challenges of Biological Data*" (2007, Bioinformatics, 23, 3343-3349).

Additional material:

Assaf Gottlieb, **Roy Varshavsky**, Michal Linial and David Horn. "*Unsupervised Feature Filtering and Instance Selection*" (Appendix).

Section 2.1

Novel Unsupervised Feature Filtering of Biological Data

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.

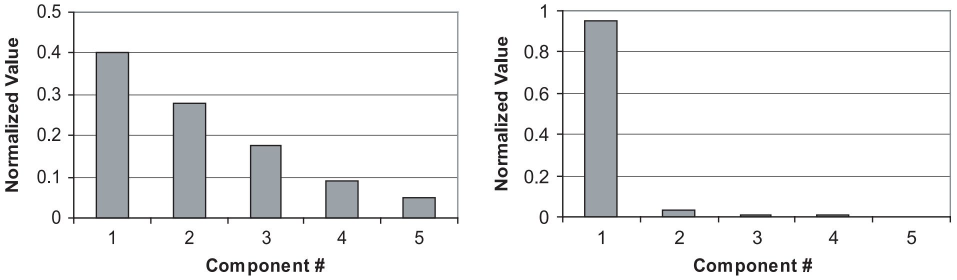


Fig. 1. A comparison of two eigenvalue distributions; the left has high entropy (0.87) and the right one has low entropy (0.14).

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] $A_{[nXm]} = \{\bar{A}_1, \bar{A}_{2,....,}\bar{A}_{i,...,}\bar{A}_n\}$, 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

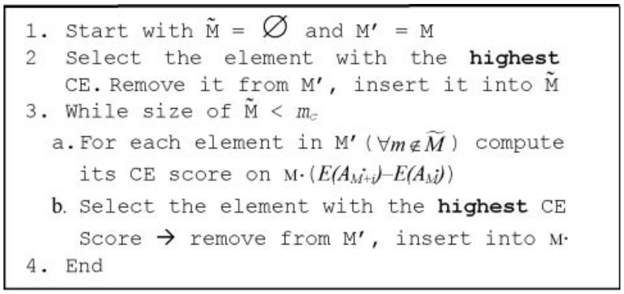
Features in the first group (high CE) lead to entropy increase; hence they are assumed to be very relevant to our problem. Retaining these features we expect the instances to be more evenly spread in the truncated SVD space. The features of the second group are neutral. Their presence or absence does 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-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

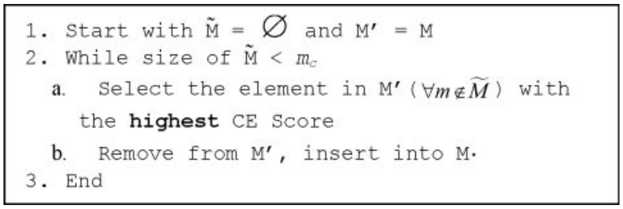
Entropy maximization can be implemented in three different ways, as is also the case in other feature selection methods.

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



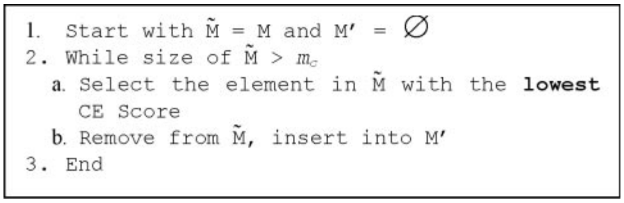
```
1. Start with $\tilde{M} = \emptyset$ and M′ = M
2  Select the element with the highest
   CE. Remove it from M′, insert it into $\tilde{M}$
3. While size of $\tilde{M} < m_c$
   a. For each element in M′ ($\forall m \notin \tilde{M}$) compute
      its CE score on M·($E(A_{\tilde{M}+i}) - E(A_{\tilde{M}})$)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

Box 1: Pseudo-code of Forward Selection method FS1



```
1. Start with $\tilde{M} = \emptyset$ 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 M·
3. End
```

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



```
1. Start with $\tilde{M}$ = M and M′ = $\emptyset$
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.

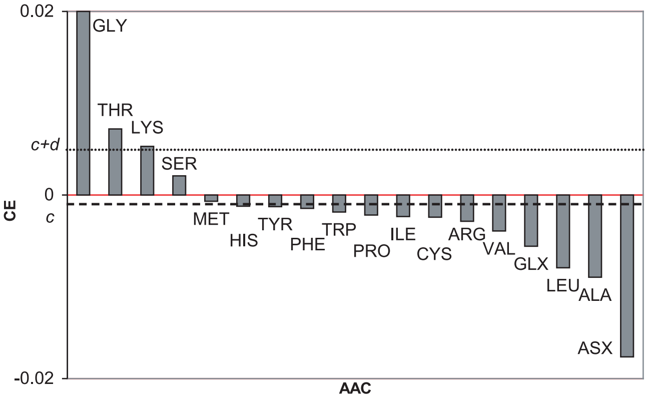


Fig. 2. CE of the 18 Amino Acid Compositions (AAC) of the virus dataset. ASX stands for ASN and ASP and GLX for GLN and GLU. The dashed line represents the value of *c* and the dot-dashed line the value of *c+d*.

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 (15th to 18th) 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

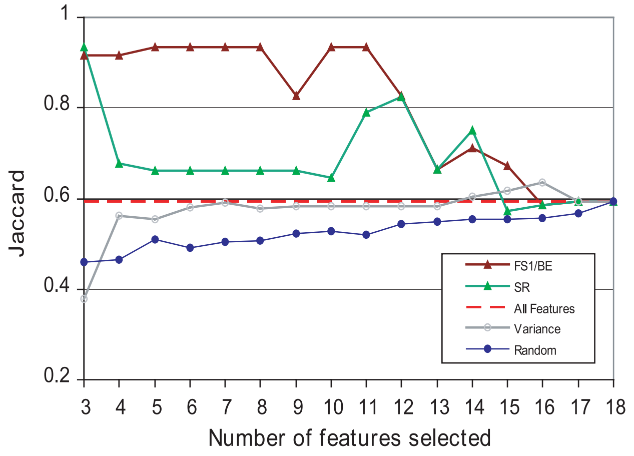


Fig. 3. Filtering quality of the virus dataset is tested by Jaccard scores of clustering performed in spaces spanned by them (see text). Best results are obtained for FS1 (identical with BE in this case) and SR for $m_c = 3$. FS1 continues to perform very well with more features. Feature selection according 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 not be fitted into this paper, can be found in the Supplementary Material[3].

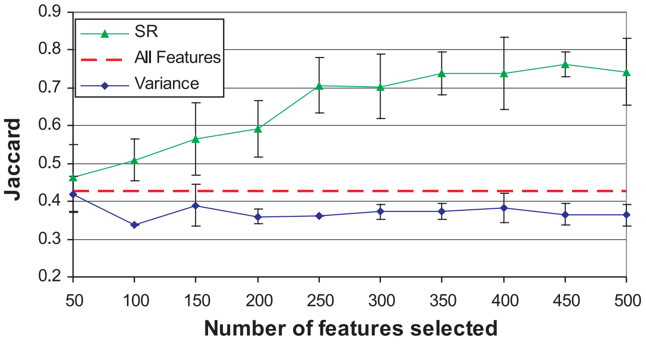


Fig. 4. Clustering quality of two feature selection methods. Results are averages of 100 runs of K-Means clustering.

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

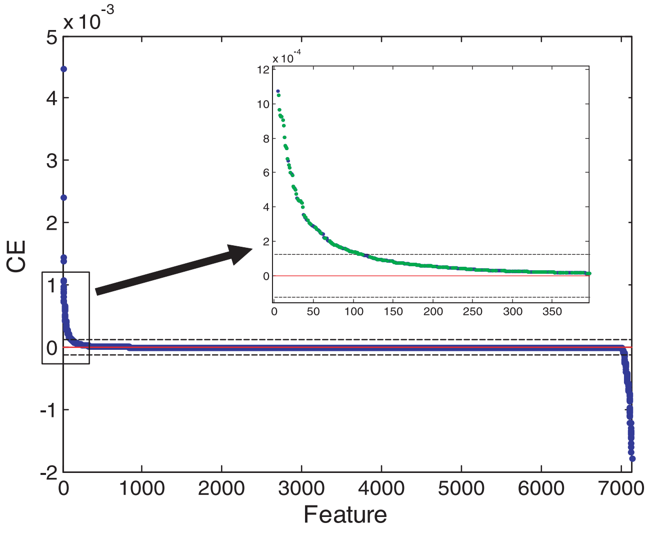


Fig. 5. CE of the 7129 genes of the Golub dataset ($c = 0$, dashed lines represent $c \pm d$). The inset zooms into the highest-ranked 300 genes, with bright dots signifying the top 100 features according to the FS1 method

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

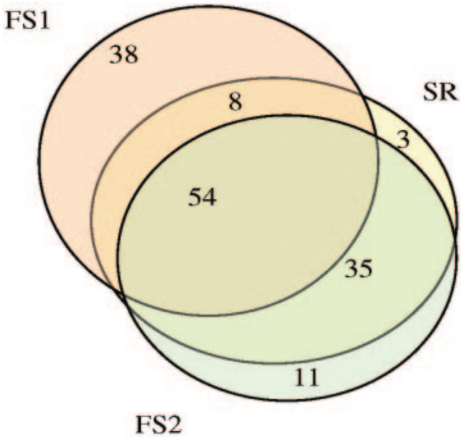


Fig. 6. Venn diagram of relations among the first 100 features selected by different methods.

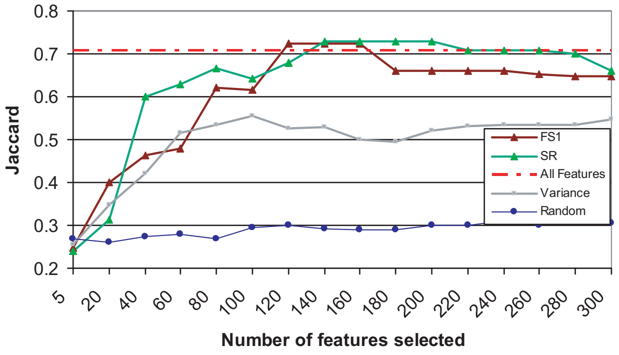


Fig. 7. Jaccard scores of QC clustering for different feature filtering methods on small gene subsets of the Golub data.

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

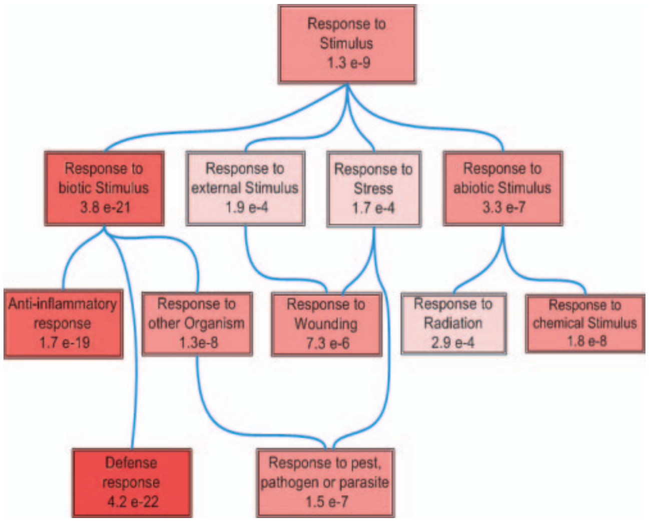

Response to Stimulus 1.3 e-9
Response to biotic Stimulus 3.8 e-21
Response to external Stimulus 1.9 e-4
Response to Stress 1.7 e-4
Response to abiotic Stimulus 3.3 e-7
Anti-inflammatory response 1.7 e-19
Response to other Organism 1.3e-8
Response to Wounding 7.3 e-6
Response to Radiation 2.9 e-4
Response to chemical Stimulus 1.8 e-8
Defense response 4.2 e-22
Response to pest, pathogen or parasite 1.5 e-7

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 $(m$-$m_c)$), leading to $O(m^5$-$(m$-$m_c)^5)$. Note that here, if $n <$ (m-m_c), the complexity is $O(mm_c n^3)$
- BE: Calculation of all CEs followed by $(m$-m_c-$1)$ repetitive diagonalization of a decreasing matrix (from m-2 to $(m_c$-$1)$), leading to $O(m^5$-$m_c^5)$. Note that here, if n < m, the complexity is reduced to $O((m^2$-$m_c^2)n^3)$.

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.

Section 2.2

Unsupervised Feature Selection under Perturbations: Meeting the Challenges of Biological Data

BIOINFORMATICS ORIGINAL PAPER

Vol. 23 no. 24 2007, pages 3343–3349
doi:10.1093/bioinformatics/btm528

Gene expression

Unsupervised feature selection under perturbations: meeting the challenges of biological data

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

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

Received on July 23, 2007; revised on September 12, 2007; accepted on October 15, 2007

Advance Access publication November 7, 2007

Associate Editor: David Rocke

ABSTRACT

Motivation: Feature selection methods aim to reduce the complexity of data and to uncover the most relevant biological variables. In reality, information in biological datasets is often incomplete as a result of untrustworthy samples and missing values. The reliability of selection methods may therefore be questioned.

Method: Information loss is incorporated into a perturbation scheme, testing which features are stable under it. This method is applied to data analysis by unsupervised feature filtering (UFF). The latter has been shown to be a very successful method in analysis of gene-expression data.

Results: We find that the UFF quality degrades smoothly with information loss. It remains successful even under substantial damage. Our method allows for selection of a best imputation method on a dataset treated by UFF. More importantly, scoring features according to their stability under information loss is shown to be correlated with biological importance in cancer studies. This scoring may lead to novel biological insights.

Contact: royke@cs.huji.ac.il

Supplementary information and code availability: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Computational biology has undergone a revolution in the last decade. One of the prominent characteristics of this revolution is the development of high-throughput technologies, allowing for gathering of large-scale data, both in the number of samples and in their features. Examples are microarray gene-expression experiments (Beer *et al.*, 2002; Khan *et al.*, 2001) and comparative genomic hybridization (CGH) (Snijders *et al.*, 2005).

A popular strategy for facilitating the analysis and interpretation of such large-scale data is selecting informative features from the thousands measured in each experiment (Guyon and Elisseeff, 2003; Herrero *et al.*, 2003). Feature selection methods are divided into two types: *supervised*, when a target function is known, and *unsupervised*, in which one has no, or limited, information regarding the samples. Supervised feature selection methods are abundant, in particular in the computational biology field, where they were found useful in improving classifications tasks (Bø and Jonassen, 2002). Nevertheless, it was argued that such methods do not lead to a unique set of selected features (Ein-Dor *et al.*, 2006). This is probably due to the fundamental variability within the data and the small number of samples (which is further reduced due to train-test partition), in comparison to the number of features.

Less studied approach is the unsupervised feature selection. Selection methods that are applied before clustering are often referred to as *filter* methods. Most methods of unsupervised feature filtering include ranking of features according to different criteria: correlation with the first principal component, range, fold-change, threshold, entropy and variance calculated on each feature individually (Guyon and Elisseeff, 2003; Herrero *et al.*, 2003). An underlying assumption for these selection methods is that only features that significantly vary along the samples carry the relevant information. Although it seems that unsupervised methods are scarce and less powerful than the supervised ones, most analysts (often inattentively), do apply some unsupervised schemes: in practice, almost every microarray analysis starts with filtering out thousands of genes with small variance or those that are below a predetermined fold-change threshold.

Recently, we have suggested an unsupervised feature filtering (UFF) framework (Varshavsky *et al.*, 2006) that was successfully applied to several datasets with various representations (e.g. gene-expression, amino-acid composition counts). UFF differs from other popular unsupervised selection schemes by (1) not involving a target function as the selection criterion [e.g. optimizing clustering results (Dy and Brodley, 2004)] and (2) considering the interplay of all features. It has been shown on several datasets of different types that a selection of only a few features according to the UFF method leads to improved clustering results relative to other unsupervised methods or to using the complete set.

Here, we investigate the effect of missing information on feature selection strategies. We employ UFF and study whether it remains valid when fractions of data are eliminated. In particular, we put emphasis on the stable features that continue to be selected under these conditions.

[*]To whom correspondence should be addressed.

Experimental data are prone to errors or information loss because of two major reasons: (i) missing or untrustworthy samples (Wang *et al*., 2006); (ii) missing values: unarguably, this is one of most bothering issues when handling gene-expression microarray datasets (de Brevern *et al*., 2004; Scheel *et al*., 2005); other microarray-based technologies (e.g. tiling array, ChIP on Chip and CGH screening) impose similar challenges. There exists a continuous drive to overcome these problems by improving the hardware (Shi *et al*., 2006), and developing imputation methods to replace missing values (Gan *et al*., 2006; Hua and Lai, 2007; Troyanskaya *et al*., 2001; Tuikkala *et al*., 2006). 'White noise' was shown to have negligible effect on the analysis (Klebanov and Yakovlev, 2007) and thus should not be considered.

Facing the fact that any data may be afflicted by missing information, we argue that a feature selection method should be relatively stable with respect to such errors. This assertion can be tested by simulating information loss and studying its effect on the method at hand. We evaluate UFF under such conditions, suggest viewing stability as a new criterion for feature selection, and study its use on biological data, leading to interesting new insights.

2 DATA AND METHODS

Figure 1 summarizes the analysis protocol. The original dataset (Section 2.1) is perturbed (Section 2.2) and filtered by UFF (Section 2.3). The selected features are then evaluated (Section 2.4) and tested with respect to their biological relevance (Section 2.5).

2.1 Datasets

A comparative analysis is performed on two (complete) gene-expression benchmarks, with known classifications, and a practical application is then applied to a Comparative Genomic Hybridization (CGH) dataset that inherently contains some missing values.

(1) SRBCT: the small round blue cell tumor gene-expression dataset includes glass-based cDNA microarray measurements of 2308 genes (features) for 83 patients (samples). The samples are categorized into four types of tumors: Burkitt lymphoma, Ewing sarcoma, Neuroblastoma and Rhabdomyosarcoma (Khan *et al*., 2001).

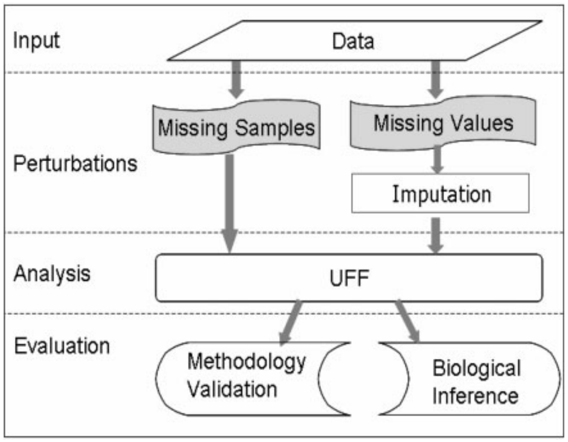


Fig. 1. Schematic representation of the analysis protocol.

(2) Lung: this HUGeneFL Affymetrix oligonucleotide gene-expression dataset (Beer *et al*., 2002), includes 86 primary lung adenocarcinomas and 10 non-neoplastic lung samples. Total 4966 genes are measured for each sample (features).

(3) CGH: this dataset (Snijders *et al*., 2005) comprises 1979 clones (features) for 89 instances (samples). The expression value of each record is the $\log_2$ratio normalized to the genome median $\log_2$ratio. The dataset contains 5807 missing values (3.3%).

2.2 Perturbations

Assuming the complete dataset is a full $[m \times n]$ matrix A, with m features describing n samples (or observations) we simulate information loss in two ways:

(1) Missing samples (Wang *et al*., 2006) are simulated by eliminating some of the columns in the matrix. We consider cases where 1%, 2%, 5%, 10%, 20% and 50% of all samples are randomly removed. Total 50 random eliminations were applied to each group size (in the leave-one-out case, all possibilities are considered).

(2) Missing values are modeled by randomly eliminating 1%, 2%, 5%, 10%, 20% and 50%, of all matrix elements. Total 50 random deletions were selected for each group size. The removed matrix elements are then imputed according to one of three imputation methods:

  (a) Standard average: each missing value is replaced with the average of all present values in the set.

  (b) Weighted average: each missing value is replaced by: [*average* (*row*) * *average* (*column*)]/*average* (*matrix*).

  (c) KNNImpute according to Troyanskaya *et al*. (2001), each missing value is replaced by the standard average of samples of the K nearest neighbors of a relevant feature ($K=10$).

For clarity, (1) description of the KNNImpute method, (2) results of 50% data loss and (3) SDs appear in Supplementary Material.

2.3 Unsupervised feature filtering (UFF)

UFF scores each one of the features according to its contribution to the SVD entropy of the dataset. Computation of the score is based on a leave-one-out principle [for a complete description see Varshavsky *et al*. (2006)].

Let A denote a matrix, whose elements A_{ij} are the measurement of feature i on sample j, e.g. expression of gene i under condition j. We base our method on the Singular Value Decomposition (SVD) procedure. It decomposes the original matrix A into $A = USV^{\mathrm{T}}$, where U and V are unitary matrices whose columns form orthonormal bases. The diagonal, non-negative matrix S is composed of singular values (s_k).ordered from highest to lowest. Let l be the rank of the matrix [$l \leq \min(m, n)$], Using the normalized relative values, ρ_k

$$\rho_k = \frac{s_k^2}{\sum_{i=1}^{l} s_i^2} \tag{1}$$

a SVD-entropy (H) can be defined (Alter *et al*., 2000):

$$H = -\frac{1}{\log(l)} \sum_{k=1}^{l} \rho_k \log(\rho_k) \tag{2}$$

SVD-entropy varies between 0 and 1. Low entropy datasets are characterized by only a few high singular values whereas the rest are significantly smaller. This pattern reflects a great redundancy in the dataset. In contrast, non-redundant datasets result in uniformity in the singular values spectrum and in high entropy.

UFF scores each feature *i* using a leave-one-out calculation of the SVD-entropy: *H* is calculated for the entire matrix and for the matrix without feature *i*. The difference in the values defines the score of feature *i*. Figure 2 displays the results after applying the UFF algorithm to the SRBCT dataset, and sorting the features according to decreasing UFF scores. Clearly, one can divide the features into three groups:

(1) Features with positive score. These features increase the entropy.

(2) Neutral features that have negligible influence on the entropy.

(3) Negative score features. These features decrease the entropy.

Note that a majority of features falls into group 2 (~92%), while groups 1 and 3 represent minorities (~4% in each). The features selected according to the UFF approach are the positive score features [lying above the threshold of mean(score) + SD(score)]. The rationale behind picking group 1 features is that, because they increase the entropy, they decrease redundancy. Hence, we may expect samples to be better separated in the space spanned by these features.

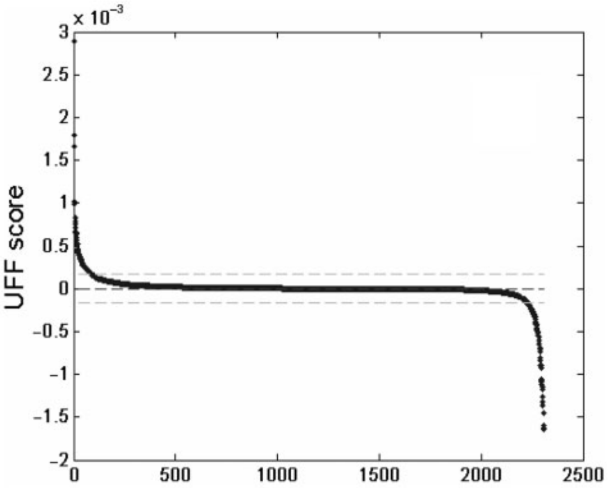


Fig. 2. UFF Scores of the 2308 genes of SRBCT features, ordered by decreasing scores. Dashed lines represent mean(score) ± SD(score).

2.4 Methodology evaluation

Given a set of selected features we evaluate it according to how successful it is in clustering correctly the set of samples, and how much it overlaps with the set of UFF selected features of the unperturbed data.

- Clustering quality. Clustering quality is measured both on perturbed and on perturbed-then-filtered datasets. Cases where the latter representation leads to higher quality indicate that the filtering is effective even though the dataset is damaged. This quality is measured using the Jaccard score: $J = n_{11}/(n_{11} + n_{10} + n_{01})$, where n_{11} is the number of pairs of samples that are classified together, both in a known classification and in the clusters obtained by the algorithm; n_{10} is the number of pairs that are classified together in the true classification, but not in the clustering and n_{01} is the number of pairs that are classified together by clustering but not in the true classification. In order to ensure that the evaluation is not biased by the clustering method, two clustering methods were compared and shown to provide consistent behavior patterns. In the two microarray datasets both QC [$\sigma = ½$, dims = 5, (Horn and Axel, 2003)] and hierarchical (Euclidian distance, average linkage) methods were considered.
- Filtering stability. Filtered features of the original and perturbed datasets are compared (Scheel *et al.*, 2005). The degree of intersection (similarity score) indicates the method's stability under the perturbation.

2.5 Stability scores

On average, each dataset has undergone ~1200 perturbations. Stability of a feature is defined as the probability of this feature to be selected under all perturbations. The features may be then ranked according to this criterion.

3 RESULTS

3.1 Methodology validation: filtering quality and stability

3.1.1 Smooth degradation of clustering quality under perturbations Figure 3 displays the clustering quality of the perturbed SRBCT and Lung datasets (missing samples and missing values with three imputation methods). UFF always

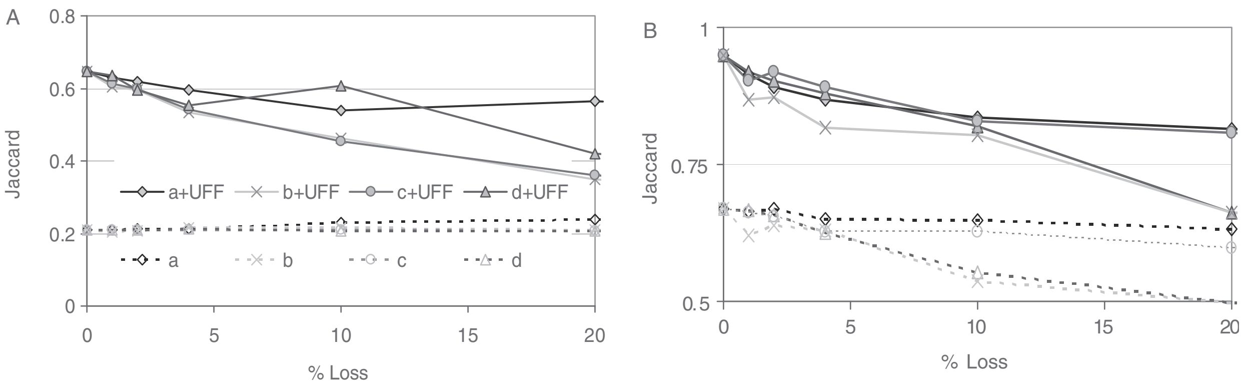


Fig. 3. Clustering results of the (**A**) SRBCT and (**B**) Lung datasets, following perturbations: missing samples (a) and missing values (with three imputation methods: (b) average, (c) weighted average and (d) KNNImpute). Dashed lines denote the clustering quality of the perturbed datasets after various levels of information loss and the continuous lined denote the corresponding quality of perturbed and then filtered sets (results shown are averages of 50 random perturbations). Detailed results for the two datasets appear in Supplementary Material.

improves clustering quality. The results degrade smoothly as a function of the amount of missing data. This allows us to draw two important conclusions: (1) UFF continues to be a good filtering method even under severe information loss. (2) There does not seem to exist a critical amount of loss beyond that clustering quality suffers a sudden drop.

In all *missing sample perturbations* cases, application of UFF improves considerably the clustering quality even under substantial information loss. This is also the case with *missing values perturbations*. Clustering after UFF outperforms clustering without UFF. Comparing between three imputation methods, we learn that the best method for the SRBCT dataset is the KNNImpute while for the Lung dataset it is the weighted average.

3.1.2 UFF is stable under perturbations The stability of filtering is measured by the similarity between the original list of features (selected when the information is complete) and the lists that are generated from the perturbed sets. The lists for the SRBCT and Lung datasets (comprising 88 and 62 genes, respectively) appear in the Supplementary Material.

Figure 4 displays the similarity scores of the perturbed SRBCT and Lung datasets as a function of the lost data. As shown, in the missing samples perturbation, the intersection levels remain high even after substantial loss. This means that UFF is stable under missing samples perturbations.

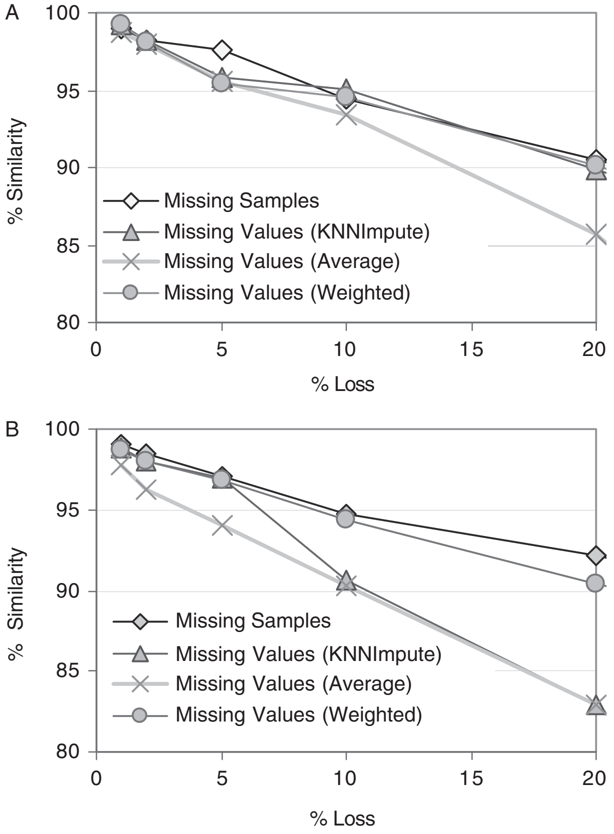


Fig. 4. Similarity levels as a function of lost data of the (**A**) SRBCT and (**B**) Lung datasets. Detailed results for the two datasets appear in Supplementary Material.

In the missing values perturbation, not all imputation methods perform equally. In both cases the simple average method performs relatively bad, while the weighted imputation method performs very well. In the SRBCT dataset the KNNimpute yields high similarity results, yet in the Lung dataset this method is found to result in less stable lists. Overall, similarity is seen to decrease linearly with information loss. In both perturbation schemes the intersection is high (~85%) even after substantial loss (20%). Similar qualitative results have been obtained by Scheel *et al.* (2005) in a supervised selection task.

3.2 Application to a faulty dataset

Given the CGH dataset that contains 3.3% missing values (see Section 2.1), we apply to it further artificial information loss in order to estimate (1) how damaging is the 3.3% original loss, and (2) which is the best imputation method.

The analysis starts with applying the three imputation methods to the dataset. Applying UFF to the three reconstructed forms, results in three lists of selected features, comprising 88, 83 and 85 clones for the average, weighted average and KNNImpute, respectively. These three lists, that are referred to as baselines, have 72 clones in common (Table S3). As shown in Figure 5, the dataset is further perturbed, both by missing values and by missing samples protocols. The resulting lists of features are then compared with their corresponding baseline lists. Figure 5 displays the

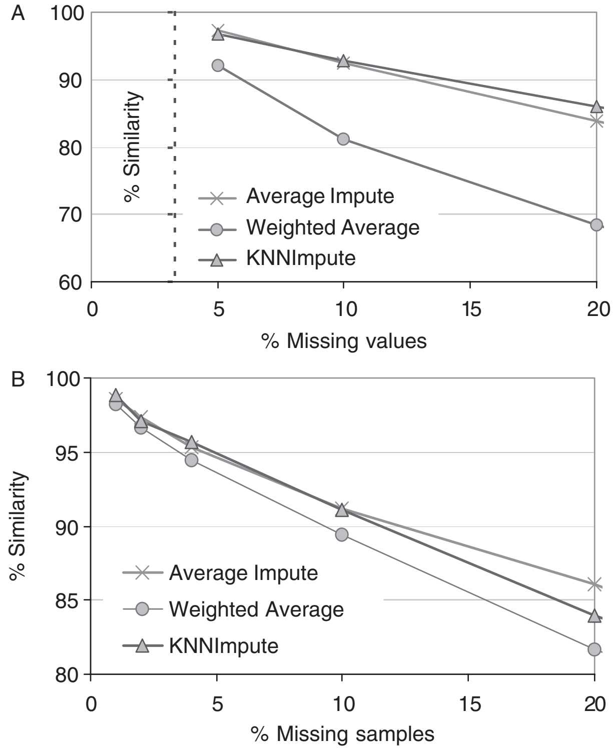


Fig. 5. Similarity scores as a function of lost data of the CGH dataset with (**A**) missing values and (**B**) missing samples perturbations. Note that the missing values analysis starts with the original 3.3% loss. Detailed results appear in Supplementary Material.

similarity scores as a function of the information loss. Note, that since three baseline lists are defined, three comparisons are applied to both protocols.

Clearly, under all perturbations, the similarity levels degrade smoothly (almost linearly), retaining high intersections (~85%) with the original lists even after substantial loss (20%). The high similarity levels may testify that, as far as clones selection is considered, the original 3.3% damage is not crucial. This observation matches the one found in the gene-expression case, which suggests that the stability characteristic of UFF is generic. Furthermore, both protocols lead to similar ranking of the different methods with weighted average inferior to the other two imputation methods.

4 BIOLOGICAL INFERENCE

In this section, we wish to study whether the stability criterion is also biologically meaningful, i.e. are the stable features causally related to the biological problem at hand?

4.1 Ranking stable features

Figure 6 displays the stability scores of the 88 first UFF genes in the SRBCT dataset (according to 0 and Varshavsky *et al.*, 2006). There exists a positive correlation between the rank order of the UFF score and stability. They are compared to the ranking of Khan *et al.* (2001) based on a supervised criterion. Out of 88, 37 of the UFF genes are common to the two lists (hypergeometric enrichment *P*-value of $1.7E^{-12}$).

Among the 10 and 20 top stable genes, 8 and 13, genes appear in the supervised-selection based list, respectively. The 20 most stable genes are listed in Table 1 (complete lists of the two datasets appear in the Supplementary Material, Tables S1A,B and S2).

4.2 Comparing stable and 'less-stable' SRBCT genes

4.2.1 Statistical analysis We conducted a statistical comparison of top 20 stable genes, with the 20 genes that were originally selected by the UFF algorithm, but found to be less stable (with stability score ranging from 0.85 to 0.51). The top stable genes have relatively low skewness and kurtosis, compared to the less stable genes. Since imputation methods tend to smooth distributions, wide symmetrical distributions should indeed be more resistant to perturbations.

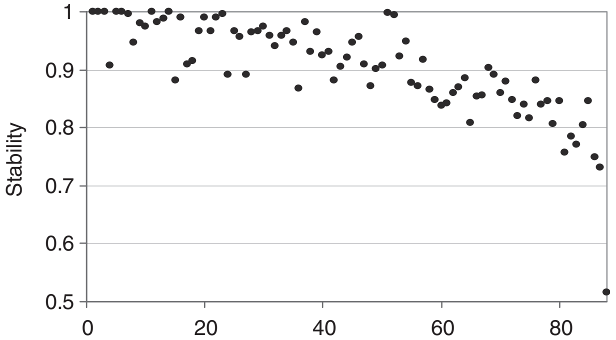


Fig. 6. Stability scores of the top scored UFF-based selection (88 genes) in the SRBCT dataset.

4.2.2 Functional analysis for the most stable genes The malignant tumors analyzed tend to occur in childhood. From a morphological view, subtle clues distinguish between the tumors. At present, analysis for chromosomal abnormalities and molecular probes are being used to assist the pathologists. The list of most stable features in the SRBCT set is intriguing. Among the top stable genes, several genes corroborate each other. Figure 7 illustrates protein–protein interactions that were experimentally validated. Several of the top 20 stable genes appear in these networks. The appearance of representative

Table 1. Top 20 stable genes in the SRBCT dataset

Stability ranking	Stability score	Genes name	UFF ranking	Khan's ranking
1–11	1	Human DNA for insulin-like growth factor II (IGF-2); exon 7 and additional ORF	1	2
1–11	1	Insulin-like growth factor 2 (somatomedin A)	2	1
1–11	1	Collagen, type III, alpha 1 (Ehlers-Danlos syndrome type IV, autosomal dominant)	3	40
1–11	1	Insulin-like growth factor binding protein 2 (36kD)	5	8
1–11	1	Human insulin-like growth factor binding protein 5 (IGFBP5) mRNA	6	62
1–11	1	SMA3	11	–
1–11	1	Actin, alpha 2, smooth muscle, aorta	14	83
1–11	1	Antigen identified by monoclonal antibodies 12E7, F21 and O13	51	73
1–11	1	IM-379708	23	–
1–11	1	Growth-associated protein 43	7	31
1–11	1	Spectrin, beta, non-erythrocytic 1	52	–
12–15	0.99	Regenerating islet-derived 1 alpha (pancreatic stone protein, pancreatic thread protein)	20	57
12–15	0.99	Nucleolin	22	–
12–15	0.99	Gelsolin (amyloidosis, Finnish type)	16	–
12–15	0.99	Troponin T2, cardiac	13	25
16–19	0.98	Crystallin, alpha B	12	79
16–19	0.98	Secreted protein, acidic, cysteine-rich (osteonectin)	37	–
16–19	0.98	Collagen, type I, alpha 2	9	–
16–19	0.98	Follicular lymphoma variant translocation 1	30	75
20	0.97	Cyclin D1 (PRAD1: parathyroid adenomatosis 1)	10	3

In addition, the ranking of the genes according to Khan *et al.* (2001) is given. '–' denote that a gene is not included in the reported 96 genes list (Khan *et al.*, 2001).

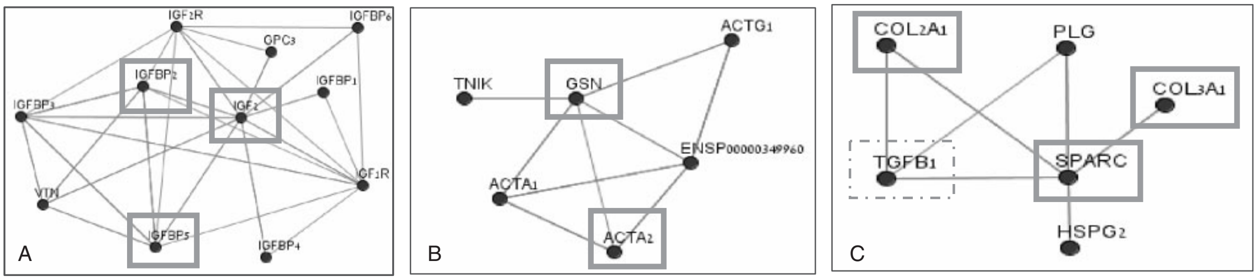


Fig. 7. Experimentally identified gene networks (von Mering *et al.*, 2007). (**A**) IGF-2 and interacting proteins; (**B**) Actins (ACT) and Gensolin (GSN) and (**C**) Collagen (COL), Osteonectin (SPARC) and TFGβ (TGFB). Genes included in Table 1 are framed (dashed frame indicates a UFF selected gene, but not among the top 20).

genes within protein networks is an indication for the importance of the identified biological process in the classification. The most evident property is that the stable genes are strongly involved in regulatory networks. In general, several genes are involved in signal transduction (i.e. IGF response), regulation of cytoskeleton and extracellular signaling.

Some genes, listed among the top ranked genes, belong to cytoskeleton elements and their regulators (including actin, gelsolin, troponin, cardiac actin alpha 2, alpha B crystalline and beta spectrin). Their roles as tumor subtype classifiers are not evident and should be experimentally validated.

The biological properties of the less stable genes are different from the top ranked 20 genes. In general, many of these genes associate with a nuclear function and thus may belong to the tumorigenesis process. Among these genes are H2A histone, DEAD/H hnRNP K, FMR1 interacting protein 2, Cyclin-dependent kinase 2-associated protein 1 and more. It is possible that they are altered in tumors, but play a weaker role in distinguishing among the different types.

5 DISCUSSION

We have subjected UFF to a perturbation-based analysis and found it to obey the condition of stability. A similar perturbation-based selection was shown to be efficient in supervised tasks (selection and classification) (Chen *et al.*, 2007). Ours is the first unsupervised perturbation-based selection procedure. We recommend using stability under perturbations as an important diagnostic tool when searching for a feature selection method.

Although for practical reasons, perturbation of even 10% should be already considered as significantly severe, in this study we extended our analysis to much higher damage levels (up to 50% of the data, see Supplementary Material). The reason for doing so is twofold: (i) acquire a deep understanding of the nature of the method and the data. It is of interest to investigate whether extensive damage, beyond some critical amount, leads to a collapse of our method (known as critical transition or percolation in various physical systems). In the problem studied here we observe a smooth, almost linear degradation in performance. (ii) In the context of gene expression, the number of unreliable or suspicious samples might often reach a significant fraction of the entire dataset. Often these samples are not literally missing but result from unreliable RNA extraction, low quality labeling, etc. We were therefore motivated to examine how removing many samples influences the lists of selected features (genes).

We have found that the effect of missing samples is very similar to the one of missing values (followed by imputation). In both, even a substantial loss of data does not significantly alter the list of the selected features, reaching a similarity of ~85%. Nevertheless, it should be emphasized that this argument should be limited to datasets with no inherent dependency among the samples. Examples for such dependencies are: time series, cell-cycle and pre-post treatment for the same individuals.

Differences in the imputation methods are identified, emphasizing that imputation method needs to be data-driven. For instance, KNNImpute is usually found perform best in the low loss region while the two average-based imputations achieve higher similarity levels at the high loss region. This last finding can be explained by the local nature of the KNNImpute method (relying only on nearest neighbors). This understanding may assist in selecting among the various imputation methods.

In the cases analyzed, a high correlation between the external and internal criteria (clustering quality and filtering stability, respectively) is reported. Specifically, in both gene-expression benchmarks the two evaluation criteria rank the imputation methods identically. This observation can be exploited to select an imputation method given a dataset. Interestingly, when applying the NRMSE (Normalized Root Mean Square Error), the standard internal criterion for evaluating imputation methods, a different methods–ranking is reported (see Supplementary Material). This suggests that our unsupervised, internal, similarity measure may be a more reliable criterion for selecting an imputation method. We therefore suggest testing the imputation method in conjunction with an unsupervised feature selection method, such as UFF. Not only does it test stability of the selected features, it also points out the best imputation method to be used under these conditions.

Identifying genes as biomarkers for tumor detection and classifications and for the multiple neurological malfunctions is of ultimate importance. Many genes selected by our stability criterion are in agreement with the ones that were found in a supervised manner. However, some potential new features are suggested. Identifying new potential markers may be due to the lack of bias in our analysis, neither from sample labeling nor from pre-selected classifier algorithm. Moreover, by applying the method on the entire dataset (without train-test splitting), we manage to reduce the well-known pitfall of over-fitting.

ACKNOWLEDGEMENTS

This research is supported by EU FR6 DIAMONDS consortium. R.V. is awarded a fellowship by the SCCB, the Sudarsky Center for Computational Biology of the Hebrew University of Jerusalem.

Conflict of Interest: none declared.

REFERENCES

Alter,O. *et al.* (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl Acad. Sci. USA*, **97**, 10101–10106.

Beer,D.G. *et al.* (2002) Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat. Med.*, **8**, 816–824.

Bø,T.H. and Jonassen,I. (2002) New feature subset selection procedures for classification of expression profiles. *Genome Biol.*, **3**, 1–17.

Chen,L. *et al.* (2007) Noise-based feature perturbation as a selection method for microarray data. In: Mandoiu,I. and Zelikovsky,A. (eds.), *ISBRA*. Springer-Verlag, Atlanta, GA, pp. 237–248.

de Brevern,A. *et al.* (2004) Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering. *BMC Bioinformatics*, **5**, 114.

Dy,J.G. and Brodley,C.E. (2004) Feature selection for unsupervised learning. *J. Mach. Learn. Res.*, **5**, 845–889.

Ein-Dor,L. *et al.* (2006) Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc. Natl Acad. Sci. USA*, **103**, 5923–5928.

Gan,X. *et al.* (2006) Microarray missing data imputation based on a set theoretic framework and biological knowledge. *Nucleic Acids Res.*, **34**, 1608–1619.

Guyon,I. and Elisseeff,A. (2003) An introduction to variable and feature selection. *J. Mach. Learn. Res.*, **3**, 1157–1182.

Herrero,J. *et al.* (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.

Hua,D. and Lai,Y. (2007) An ensemble approach to microarray data-based gene prioritization after missing value imputation. *Bioinformatics*, **23**, 747–754.

Khan,J. *et al.* (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.*, **7**, 673–679.

Klebanov,L. and Yakovlev,A. (2007) How high is the level of technical noise in microarray data? *Biol. Direct*, **2**, 9.

Scheel,I. *et al.* (2005) The influence of missing value imputation on detection of differentially expressed genes from microarray data. *Bioinformatics*, **21**, 4272–4279.

Shi,L. *et al.* (2006) The microarray quality control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nat. Biotechnol.*, **24**, 1151–1161.

Snijders,A.M. *et al.* (2005) Rare amplicons implicate frequent deregulation of cell fate specification pathways in oral squamous cell carcinoma. *Oncogene*, **24**, 4232–4242.

Troyanskaya,O. *et al.* (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.

Tuikkala,J. *et al.* (2006) Improving missing value estimation in microarray data with gene ontology. *Bioinformatics*, **22**, 566–572.

Varshavsky,R. *et al.* (2006) Novel unsupervised feature filtering of biological data. *Bioinformatics*, **22**, e507–e513.

Mering,C. *et al.* (2007) STRING 7 – recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res.*, **35**, D358–362.

Wang,D. *et al.* (2006) Effects of replacing the unreliable cDNA microarray measurements on the disease classification based on gene-expression profiles and functional modules. *Bioinformatics*, **22**, 2883–2889.

Chapter 3

Global Considerations in Hierarchical Clustering Reveal Meaningful Patterns in Data

Abstract[*]

Background: A hierarchy, characterized by tree-like relationships, is a natural method of organizing data in various domains. When considering an unsupervised machine learning routine, such as clustering, a bottom-up hierarchical (BU, agglomerative) algorithm is used as a default and is often the only method applied.

Methodology/Principal Findings: We show that hierarchical clustering that involve global considerations, such as top-down (TD, divisive), or glocal (global-local) algorithms are better suited to reveal meaningful patterns in the data. This is demonstrated, by testing the correspondence between the results of several algorithms (TD, glocal and BU) and the correct annotations provided by experts. The correspondence was tested in multiple domains including gene expression experiments, stock trade records and functional protein families. The performance of each of the algorithms is evaluated by statistical criteria that are assigned to clusters (nodes of the hierarchy tree) based on expert-labeled data. Whereas TD algorithms perform better on global patterns, BU algorithms perform well and are advantageous when finer granularity of the data is sought. In addition, a novel TD algorithm that is based on genuine density of the data points is presented and is shown to outperform other divisive and agglomerative methods.

Application of the algorithm to more than 500 protein sequences belonging to ion-channels illustrates the potential of the method for inferring overlooked functional

[*] A preliminary version of this study (**Roy Varshavsky**, David Horn and Michal Linial. "*Recursive Top-Down Quantum Clustering of Biological Data*") was orally presented at ISMB, PLoS Track, 2006.

annotations. ClustTree, a graphical Matlab toolbox for applying various hierarchical clustering algorithms and testing their quality is made available.

Conclusions: Although currently rarely used, global approaches, in particular, TD or glocal algorithms, should be considered in the exploratory process of clustering. In general, applying unsupervised clustering methods can leverage the quality of manually-created mapping of proteins families. As demonstrated, it can also provide insights in erroneous and missed annotations.

Introduction

Clustering is a common unsupervised machine learning procedure. It is often used for preprocessing, and usually provides a general overview, especially when dealing with large datasets. Its applications range from astronomy to economics, psychology marketing, text mining and other areas. Recent advances in genomic biology high-throughput techniques have led to a growing need for efficient and powerful clustering algorithms (D'Haeseleer, 2005). For instance, in large-scale gene expression data, clustering algorithms are useful in the diagnosis of different samples (e.g., diseased and healthy patients, labeling of tissues by disease subtype), as well as for their ability to reveal functional classes of genes among the thousands often used in experimental settings (Eisen, et al., 1998; D'Haeseleer, 2005).

Clustering algorithms are often classified as either nonhierarchical (partitioning) or hierarchical. The former define a complete partition of the data (for comprehensive reviews see (Jain and Dubes, 1988; Duda, et al., 2000; D'Haeseleer, 2005). Because they suggest multiple levels of organization, hierarchical algorithms are best suited for describing data that have some inherent breakdown resolution. Organizing complex arrangements into hierarchies is a common technique in many fields, such as grammar description in computational linguistics, industrial organization (NAICS - The North American Industry Classification), object oriented programming, biological taxonomy and evolutionary organization of proteins, genes or species. Hierarchical clustering has been successfully applied to protein sequences, chemical entities, 3D structural information and protein catalytic activities (Handl, et al., 2005).

The outcomes of hierarchical algorithms can be represented as a tree, where each node branches into two (a 'binary tree') or more nodes. Ideally, the tree has some underlying basis; for instance, sub-industry breakdown, or protein families that reflect evolutionary diversification. In any case, it can represent many clustering solutions corresponding to different groupings of nodes. A collection of nodes may be viewed as natural cuts in the tree. Some of the clustering possibilities may match an expert's view. Other clusters may correspond to a pattern exposing the nesting in the data (sub-classes) which a given expert may not have been aware of. In fact, this is the rationale behind the clustering approach; namely, finding new internal patterns in the data. Since hierarchical clustering provides alternative clustering possibilities, it is usually considered as a richer tool than the single, nonhierarchical, clustering solution.

Hierarchical methods can be further divided into Bottom-Up (BU, agglomerative) and Top-Down (TD, divisive) types (Jain and Dubes, 1988; Duda, et al., 2000; Planet, et al., 2001). BU algorithms start with each instance as a cluster and repeatedly merge clusters until a unified cluster is formed. They are popular in genomics (gene expression, (D'Haeseleer, 2005) and proteomics (Rune, 2007), and have been implemented in resources such as ClusTr (Apweiler, et al., 2001) and ProtoNet (Sasson, et al., 2003). TD methods work in the opposite direction and are rarely used for these types of data. Although most tutorials present the two strategies, and some works have recently suggested ways to combine them (Chipman and Tibshirani, 2006), BU algorithms are significantly more popular than TD algorithms. A survey of all articles published in PLoS in the last two years (years 2006-2007) shows that out of 86 publications that apply hierarchical clustering to analyze data, only 3 do not utilize the standard BU approach. This significant bias toward the BU approach is mostly due to its availability in software packages (Eisen, et al., 1998; MathWorld) and intuitive appeal. Furthermore, the reliability at the beginning of the clustering process is evident and no assumption on any statistical model in the data is required. These reasons probably led most researchers to neglect the TD approach as a potential approach for unlabeled data.

Although less popular, several recent TD algorithms have been found to be highly efficient, especially in document classification problems. One such example is the Bisecting K-Means algorithm, based on the divide-and-conquer scheme of repeated K-means (K=2). It outperforms both standard K-Means and agglomerative clustering (Steinbach, et al., 2000), and is computationally efficient (Cimiano, et al., 2004). It suffers, however, from the usual problems of the K-means approach; namely a bias toward spherical clusters and a dependency on initial conditions. The second such example is Principal Direction Divisive Partitioning (PDDP), which is based on repeated divisions of instances according to the sign of their projection on the first principal component (Boley, 1998). PDDP outperforms the bisecting K-Means algorithm in quality and stability (Savaresi and Boley, 2004) and will thus be used here as a benchmark for a state-of-the-art TD algorithm.

This paper examines the advantages of involving global approaches in clustering, and demonstrates that they can generate meaningful results near the top of the hierarchy tree. It tests and compares different approaches on three extensively studied benchmarks. The TD algorithms succeed better in capturing the expert assignment as compared to the state-of the-art BU clustering methods. Moreover, a novel TD algorithm, called TDQC (Top-Down Quantum Clustering) is then presented and shown to outperform other algorithms. TDQC is based on an algorithm which has been applied to gene expression datasets (Varshavsky, et al., 2005) that were initially processed by SVD. In addition, an intermediate approach, named ‘glocal’, which is a BU based clustering with global consideration, is suggested to handle datasets represented by distances (and not in their feature space).

The datasets and the algorithm are described in the next section. After the comparative study of various TD and BU algorithms on the three benchmarks we apply them to a functionally coherent protein dataset. The application of TD to a protein set leads to biological insights that can reveal intriguing patterns in the data. ClusTree, a new validation and visualization tool that was used to compare the performance of the different hierarchical classification methods is provided.

Materials and Methods

Datasets

Various clustering methods are applied to four different types of datasets. These sets are the basis for a comparative analysis of previous studies and existing algorithms. Two of the sets are known benchmarks of gene-expression experiments. The third set is a known stock-market dataset, and forth is a biological dataset of ion-channel proteins.

Cell Cycle genes Spellman *et al.* identified 798 genes as cell cycle regulated in the yeast *Saccharomyces cerevisiae* and catalogued them into five classes that correspond to different stages of the yeast cell cycle (marked as M/G1, G1, S, G2 and M). Expression levels of those genes were recorded at 72 continuous time-points yielding a [798 genes x72 time-points] matrix.

Leukemia patients The Golub *et al.* dataset has served as a benchmark for several clustering methods (Golub, et al., 1999; Getz, et al., 2000; Sharan and Shamir, 2000). The experiment sampled 72 patients with two types of leukemia, ALL and AML. The ALL set is further divided into T-cell and B-cell leukemia and the AML set is divided into patients who underwent treatment and those who did not. For each patient, the expression levels of 7129 genes is reported. The clustering task is to find the four cancer groups within the 72 patients in a [72 patients x7129 genes] gene expression matrix.

Standard and Poor (S&P) We used the stocks dataset of (Slonim, et al., 2005), who collected day-to-day fractional changes in the price of all stocks in the Standard and Poor's 500 list during the 273 trading days of one year. 487 of the stocks are divided in 10 different industry segmentations. The dataset is organized in a [487 stocks X 273 trade days] matrix.

Ion Channel proteins: The dataset is extracted from the SwissProt database (version 40.28). For the 614 proteins that are annotated as 'ion channel activity' (according to Gene Ontology, ID-5126), all-against-all BLAST E-values are recorded (Altschul, et al., 1997). All E-values lower than 100 are kept in a matrix and E-values higher than 100 are limited to be 100. 518 of these proteins are annotated by the InterPro (http://www.ebi.ac.uk/interpro/, version 7.0) collection, thus resulting a [518 proteins x 518 proteins] distances matrix. Only exclusive InterPro labels were considered. There are

~40 exclusive InterPro labels that are associated with at least 2 proteins each. Several levels of granularity are associated with this protein set. The 3 group labels are 'ligand-gated channel', 'voltage gated' and 'others'. These 3 classes describe a gross partition. This gross classification can be nested into 11 classes which can be further nested into 19 classes. The 3 resolution levels are considered *gross*, *medium* and *detailed* mapping (Table 1, supplementary material).

The TDQC algorithm

The TDQC algorithm is defined in Box 1:

```
TDQC Algorithm:
0. Define original dataset (Number of sets = 1)
1. [Optional] Apply preprocessing to each set
2. Run QC (Quantum Clustering) on each set
3. Divide each set into two sets containing:
   a. Instances belonging to the cluster with the global
      minimum (A in Fig. 1)
   b.  All the rest (B in Fig. 1)
4. Recursively go-to 1 for each set including more than 2
   instances
```

Preprocessing

In order to transform the data into a compressed, manageable and hopefully noise-free representation, it is recommended to use the Singular Value Decomposition (SVD) method. SVD represents any real matrix X of size *[nXm]* as a product $X=U\Sigma V^T$, where U and V are orthonormal matrices and Σ is a diagonal matrix whose eigenvalues s_i (singular values) appear in decreasing order. In this context, n is the number of instances (or elements), and m is the number of features (or attributes), describing each instance. The columns of U and V define two independent vector spaces. Rather than studying the resulting low-rank matrix $X'=U\Sigma' V^T$ (by zeroing all singular values at locations $i>r$, one can compress the data into an r-dimensional space), we focus our attention on the r first columns of the unitary matrices U and V. It is within these vector spaces that we look for cluster structures (Alter, et al., 2000; Horn and Axel, 2003; Varshavsky, et al., 2005).

Following the experience of Latent Semantic Analysis (LSA), in computation linguistic (Landauer, et al., 1998), we define distances among the *r*-dimensional vectors in terms of cosines of the angles among them, as *d*=1-cos(Θ).

Quantum Clustering (QC)

The Quantum Clustering (QC) algorithm (Horn and Gottlieb, 2002) begins with a Parzen window approach, assigning a Gaussian of width σ to each data-point, thereby constructing *Ψ(x)*, where

$$\psi(x) = \sum_{i=1}^{N} e^{-\frac{(x-x_i)^2}{2\sigma^2}} .$$

Ψ(x) can serve as a probability density that could have generated the data. Assuming this function to be the ground-state (lowest eigenvalue) of the Hamiltonian H of the Schrödinger equation:

$$H\psi = (-\frac{\sigma^2}{2}\nabla^2 + V(x))\psi(x) = E\psi(x),$$

one can solve for the potential energy V uniquely, determining E through the condition that min(V(x))=0. The Schrödinger equation can be understood as a model balancing a clustering force (represented by the potential V) and a dispersive force (the second derivative term), that it is responsible for the fact that the data are not concentrated at the minima of V (bottoms of the potential energy).

An example of V(x) is shown in Fig. 1 for a dataset that comprises 798 genes. The classification of the genes into phases of the cell-cycle is illustrated by the different colors. The original data are given in 72 dimensions (time points). SVD is used to reduce them to two dimensions. The x-axis of this figure corresponds to cos(Θ) of each of the 2D vectors representing the genes. As Fig.1 displays a cyclic trend is well observed. In conventional QC one would cluster the instances according to the valleys of V that they belong to. In TDQC we separate the data into two sets, α and β, where set α is defined by the deepest valley of V. To each dataset we reapply preprocessing, QC and division in a recursive manner. The stopping criterion of the recursion is when a subset contains no more than 2 data points. It is noteworthy that although SVD preprocessing is not a mandatory step, according to our experience, this routine is found very effective in both improving the clustering results and in significantly reducing the algorithm's runtime.

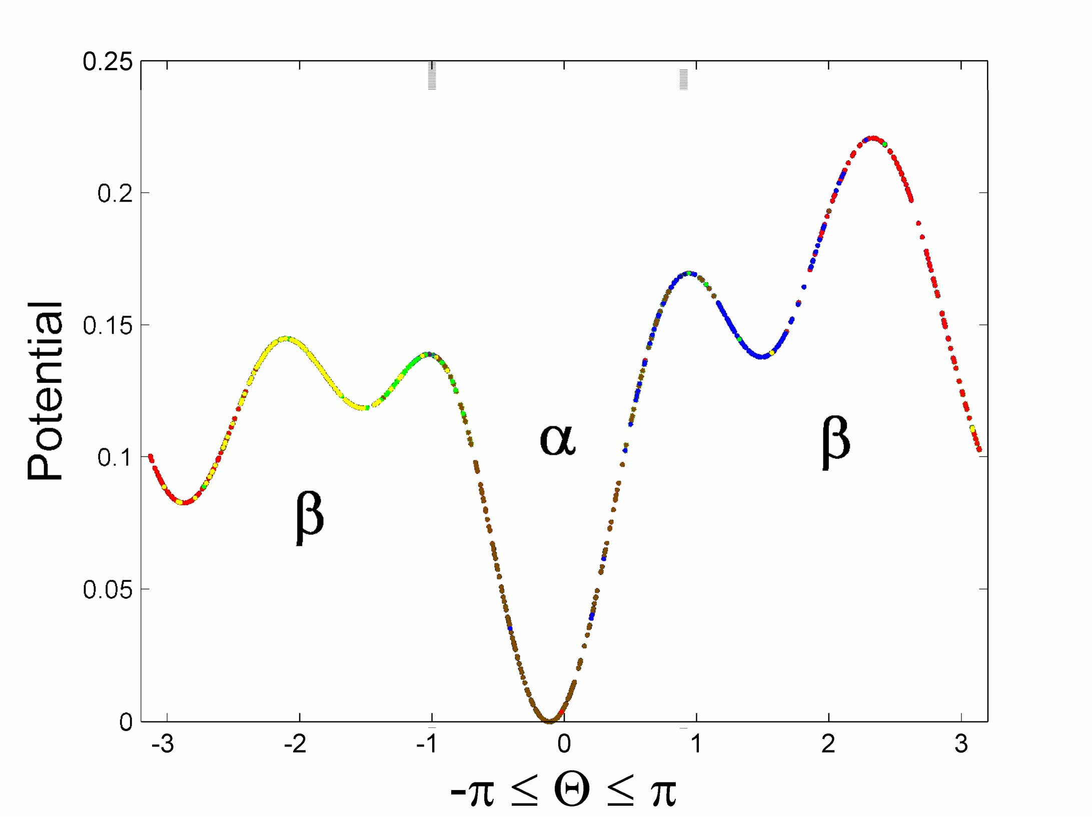


Fig 1. Potential values of the cell cycle dataset. Data were projected onto the two leading SVD components, and represented in terms of the angle in these coordinates. Dashed lines mark the partitioning of the dataset into two groups (α and β). For details see text. The color code represents Spellman's expert view for the 5 cell cycle phase (G1-brown, number of instances, S-green, S/G2- yellow, G2/M-red, M/G1-blue).

'Glocal' Hierarchical Approach: considering global information in bottom-up clustering

Data may come in two possible representations: *(1)* Feature space (a [*nXm*] matrix): each instance is measured according to its features (or attributes). Examples are: Gene expression, 3D coordinates of protein structures. *(2)* Distances or similarities (a [*nXn*] matrix): each instance is presented by its distance or similarity to another instance. Examples are: BLAST or Smith-Waterman matrices in proteomics. This representation leads to a square, and often, symmetric matrix.

Clearly, the second representation is less informative than the first. It can be calculated from the first but, given only the distances, feature space cannot be reconstructed (except approximately as in Multidimensional Scaling (Kruskal and Wish, 1981)), as shown in Fig. 2 (A, and B). Standard BU relies on distances only, even when the data are given in feature space (e.g., in gene-expression analysis): distances are first derived and iterative lineage is performed on them (Planet, et al., 2001).

In the cases where data is represented only by distances (Fig 2B), we argue that considering only the 'nearest neighbors' as the standard BU algorithms suggest, might end up neglecting relevant information in the data. We therefore suggest adding a global perspective to local clustering, namely glocal (global-local) clustering. This may be achieved by treating the distance matrix as an instance-by-feature matrix, i.e. using the instances as defining feature-space, after which BU is applied (Fig 2C). The instance-by-feature matrix allows one also to apply processing routines (e.g., SVD, PCA) to achieve dimensionality reduction before applying the clustering algorithm (see, e.g., Varshavsky, et al., 2005).

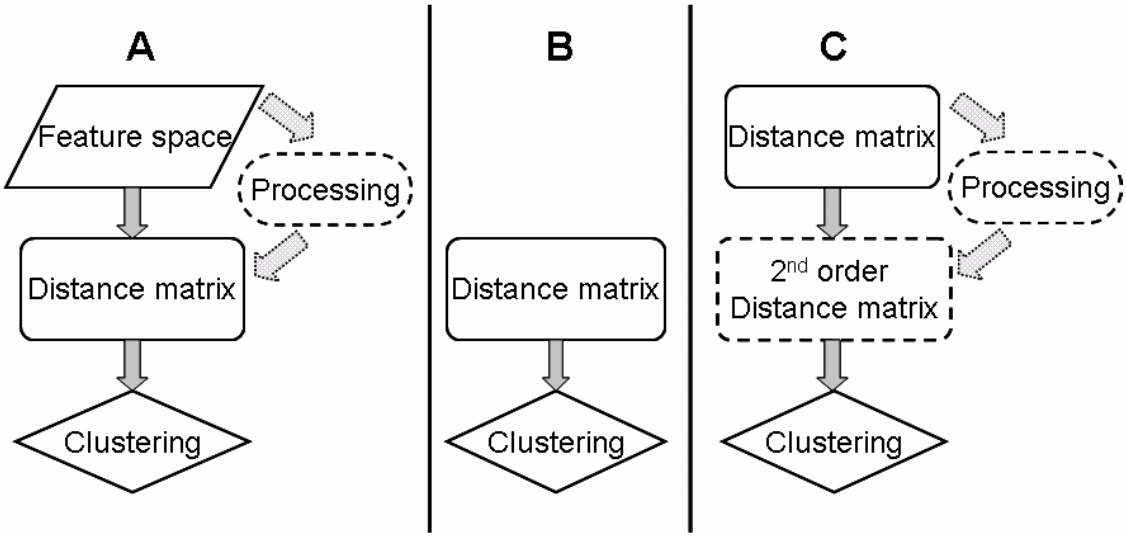


Fig. 2. Three possible ways to handle data for generalized BU clustering: (A). Standard workflow when data are presented in feature space. (B). Standard workflow when only the distance matrix is known. (C). Our 'glocal' algorithm manipulates the distance matrix by using feature-space methods. Light gray arrows denote optional steps and dotted frames denote global consideration, such as SVD or PCA manipulations.

Statistical Criteria for Classification Quality

A clear limitation of hierarchical clustering (whether TD or BU) is the inherent difficulty in the evaluation scheme. Jain & Dubes argue that the hierarchy of clustering can be evaluated only when an expert-hierarchy is available (we use the term 'expert' to describe the external data labeling (Jain and Dubes, 1988)). Quite often such expert-hierarchies are unavailable and no gold standard criterion exists (Cimiano, et al., 2004). Alternative measures that do not capture the hierarcy per-se have been suggested (Torrente, et al., 2005).

We address the instances where expert-classification of data is provided, and combine 3 assessment methods to describe different qualities of the clustering tree.

1. **Node Score** Since each node specifies a cluster, enrichment *p*-values can be calculated to assign the given node with one of the classes in the data. This is done by using the hypergeometric probability density function. The significance *p*-value of observing *k* instances assigned by the algorithm to a given category in a set of *n* instances is given by $p = \sum_{x=k}^{n} \binom{K}{x}\binom{N-K}{n-x} / \binom{N}{n}$, where *K* is the total number of instances assigned to the class (the category) and *N* is the number of instances in the dataset. The *p*-values for all nodes and all classes may be viewed as dependent set estimations; hence we apply the False Discovery Rate (FDR) criterion to them requiring $q<0.05$ (Benjamini and Hochberg, 1995). *P*-values that do not pass this criterion are considered non-significant. We further apply another conservative criterion; namely, a node is considered significant only if $k \geq n/2$ (i.e., the majority of its instances belongs to the enriched category).
2. **Level Score** A level *l* of the tree contains all nodes that are separated by *l* edges from the root, i.e., that share the same Breadth First Search (BFS) mapping. Each level specifies a partition of the data into clusters. Choosing for each node, the class for which it turned out to have a significant node score, we evaluate its Jaccard-score *(J=tp/(tp+fn+fp),* where *tp* is the number of true positive cases, *fn* the number of false negative cases and *fp* the number of false positive cases) . If the node in question has been judged to be non-significant by the enrichment criterion, its J-score is set to null. The level score is defined as the average of all J-scores at the given level.

3. **Tree Score** We define the weighted best-J-Score ($J^{*} = \frac{1}{N}\sum_{i}^{c} n_i J_i^{*}$) where J^{*}_{i} is the best J-Score for class i in the tree, n_i is the number of instances in class i, c is the number of classes and N is the number of instances in the dataset. This criterion provides a single number specifying the quality of the tree based on a few nodes that contain optimal clusters. This score or its close variation has been applied to measure the quality of proteins families (Kaplan, et al., 2004) and document classification (Steinbach, et al., 2000; Zhao and Karypis, 2002).

Results

All datasets were analyzed using two nonhierarchical algorithms, QC and K-Means, several variants of Bottom-Up algorithms, single-linked (BU-S), average-linked (BU-A) and complete-linked (BU-C) (Jain and Dubes, 1988; Duda, et al., 2000) and two Top-Down algorithms, PDDP and our TDQC.

The results of the hierarchical algorithms were evaluated using a combination of the 3 scoring methods presented above as follows. *(A)* The node-score, the clustering tree is presented with its enrichment markers for every tree node. It combines a qualitative and graphical description of the results. Recall that the graphical description is presented for visualization purposes only. *(B)* The level-score, the average J-score of each level in the tree, which provides both qualitative and quantitative information on the algorithm's performance along the hierarchy. *(C)* The tree score, the weighted best J-scores. Being a single score, the tree score provides a criterion for comparison of hierarchical algorithms to algorithms that are nonhierarchical in nature.

Fig. 3 A, B displays the trees as generated by a BU-A algorithm (using Euclidean metric and average linkage), and the TDQC algorithm when applied to the Cell Cycle dataset. Note that the BU-A performed best out of all the BU variants (Table 1).

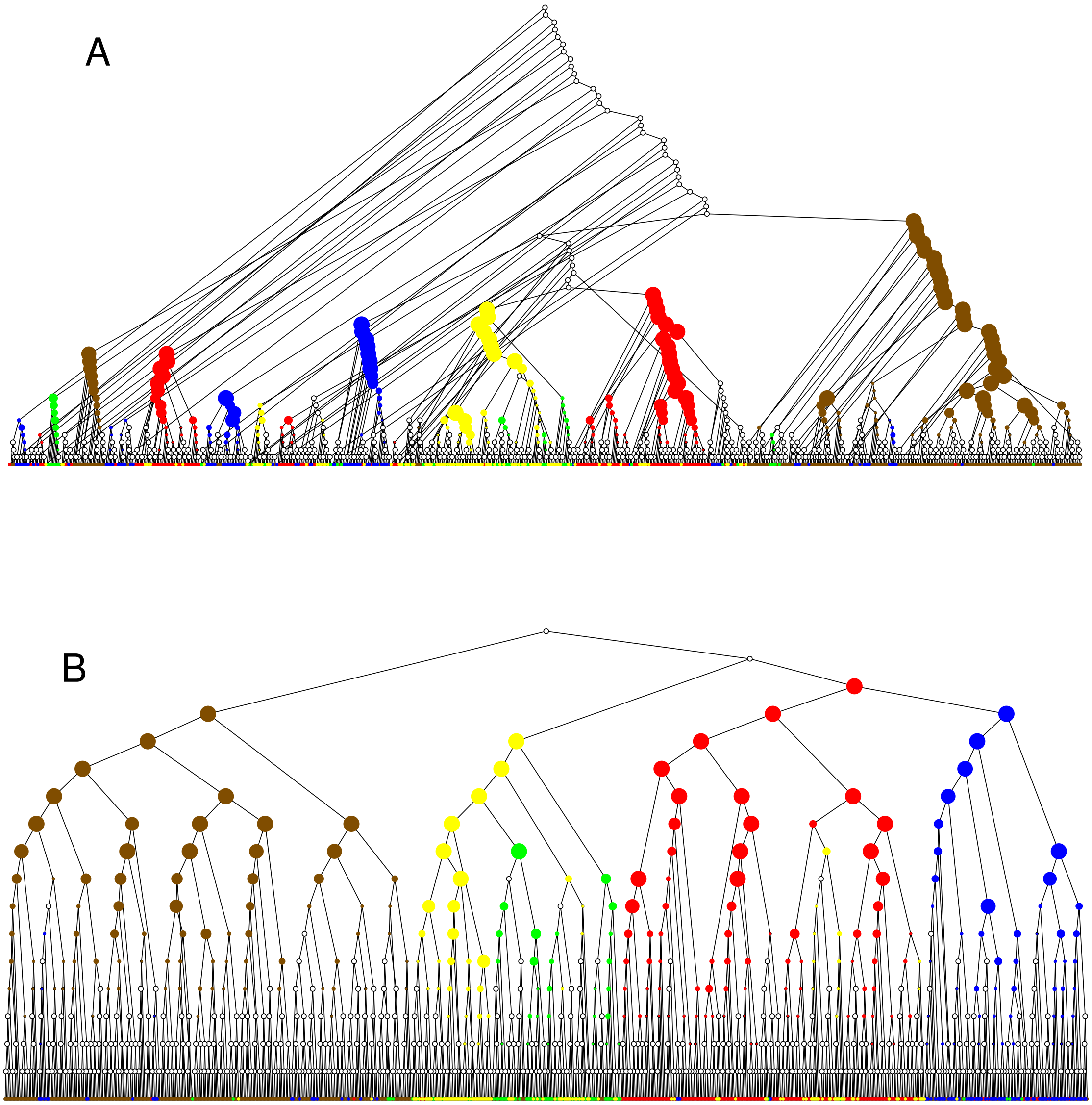


Fig 3. Hierarchical trees of the 798 cell cycle genes for BU-A (A) and TDQC (B) algorithms. Color codes specify the five cell cycle classes as in (Fig. 1). Dot sizes indicate statistical enrichment levels (larger sizes correspond to smaller p-values). Uncolored nodes represent non-significant enrichment.

Some prominent patterns emerge from Fig. 3 A, B and almost identical conclusions can be drawn from all other datasets: *1.* The BU tree is far more unbalanced relative to the TD tree. *2.* The TD algorithm performs best on higher levels of the tree, whereas the BU algorithm performs better on lower levels of the tree. This can be seen here by observing where the statistical enrichment of nodes is highest. *3.* TD clusters (sub-trees) are very coherent, i.e. it is very rare for significant nodes of one color to have children of another color.

Next we turn to measuring clustering quality by comparing level-scores in Fig 4. The TDQC algorithm has a high maximal score (0.44) and displays an almost monotonic decrease with increasing tree-level. The BU algorithm exhibits significantly different behavior. Namely, it leads to a bimodal distribution and its much smaller (0.13) maximal score is located at low hierarchy levels.

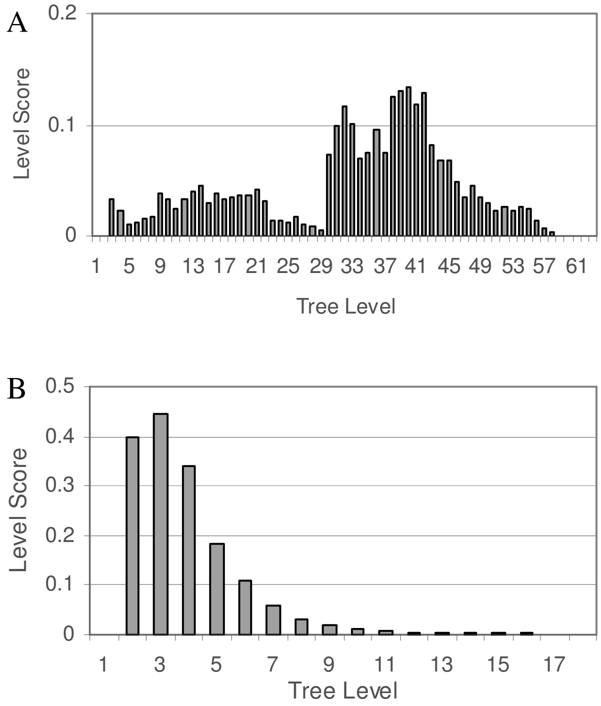


Fig. 4. Level scores of (A) BU-A and (B) TDQC for the cell cycle dataset. Tree levels are counted from the root. Note the different scale for the Y-axes.

The two trees also differ in their tree depth. The depth of the tree (*D*) is defined as the distance between the root and the farthest leaf. A completely balanced (binary) tree with *N* nodes is $log_2(N)$ deep whereas a completely unbalanced tree is *N* deep. Figure 5 displays the relative depths ($(D- log_2(N))/(N- log_2(N))$) of all trees generated by different BU and TD algorithms when applying them to the 4 datasets presented in this study.

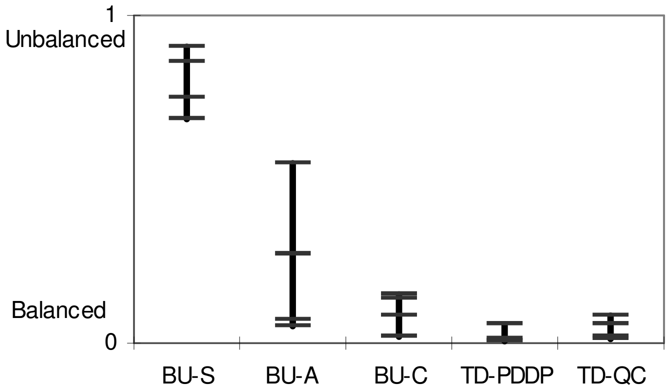


Fig. 5. The relative depths of the trees generated by the various algorithms when applied to 4 gene expression, stock market and protein family.

Despite the fact that each of the datasets used in this study comprises a different number of instances and is differently represented (e.g., similarity, raw data), we observe common trends in Fig. 5 and conclude that the nature of the algorithm governs the shape of the tree. TD algorithms tend to generate more balanced trees, and as a result have fewer levels (in the PDDP algorithm each binary division is essentially into sub-clusters of equal sizes); BU algorithms usually generate deeper trees where single-linked (BU-S) algorithms tend to produce chain-like trees, whereas complete-linked algorithms (BU-C) create more balanced trees (Hansen and Delattre, 1978)

Finally we turn to the global measures of clustering quality, based on comparisons with expert classifications. Table 1 summarizes the tree scores of all algorithms when applied to the gene-expression and the stock-market benchmarks. The TD algorithms outperform the BUs in all these cases. This is presumably due to the fact that the expert classifications represent global partitions of the data, whereas the BU approaches are fairly poor (BU-S in particular, D'Haeseleer, 2005). TDQC outperforms all other algorithms, including the nonhierarchical QC.

	Elements	Features	Classes	Non-hierarchical		Hierarchical				
						BU			TD	
				QC	K-Means	BU-S	BU-A	BU-C	PDDP	TDQC
Cell cycle	798	72	5	0.613	0.537 (0.06)	0.265	0.472	0.409	0.542	**0.646**
Leukemia	72	7129	4	0.758	0.519 (0.1)	0.465	0.522	0.53	0.545	**0.804**
S&P	487	273	10	0.400	0.306 (0.05)	0.2	0.261	0.445	0.441	**0.504**

Table 1. Clustering scores (tree score) of nonhierarchical (QC, K-Means) and hierarchical algorithms. K-Mean was performed 10 times and averaged (and std is in parenthesis), Hierarchical algorithms are BU (S, A and C marks the Single, Average, Complete, respectively) and TD (PDDP, TDQC) algorithms. Best scores are bold faced.

Evaluation of Different Granularity Levels in Protein Sets

In order to expand our analysis on data that are inherently hierarchical, we analyzed a set of proteins associated with annotation of channels. This set comprises well-studied proteins to which functional annotations are assigned based on experimental evidence and evolutionary homology relationships (Ren and Paulsen, 2005). Our set is composed of proteins associated with ‘ion channel activity’, which form a subset of proteins belonging to ‘transporters and channels’ (Gene Ontology ID-6811). These are membranous proteins that function in the directional translocation of substances across membranes. The directional translocation is between cell compartments and between cells and the environment. These proteins are defined by InterPro experts as belonging to 3 classes according to their gating mode: ligand-gated, voltage-gated and ‘others’. The last group includes proteins that are gated by nucleotides (e.g., as in the case of the cystic fibrosis chloride channel) and several channels that have a mixed gating mode or yet undefined properties. This ‘gating mode’ property dominates other characteristics of the channels and receptors including their multimeric nature, the number and nature of their accessory subunits, the number of transmembrane domains, etc. These 3 classes are further divided into other granularity levels of 11 and 19 classes respectively (see Methods).

We tested the various clustering algorithms to see how well they met the different granularity levels (Table 2). This served to show which approaches are appropriate at different granularity levels.

Classes	Non-hierarchical		Hierarchical				
			BU			TD	
	QC	K-Means	BU-S	BU-A	BU-C	PDDP	TDQC
3	0.6859	0.565 (0.13)	0.613	0.395	0.382	0.771	**0.808**
11	0.4626	0.533 (0.05)	0.338	0.34	0.245	0.567	**0.61**
19	0.3218	0.515 (0.06)	0.23	0.32	0.268	0.64	**0.655**

Table 2. Clustering scores of different algorithms applied to the ion channel proteins. Scores are measured according to the appropriate granularity level (for 3, 11 and 19 classes).

Clearly, comparing the performance of the different algorithms for different granularity levels (Table 2) shows the inferior performance of the BU algorithms. To address the question of suitability of the algorithm to the data, we compared the best TD to the best BU algorithms (TDQC and BU-A, respectively). Since the BU level-scores have a bimodal pattern (as in Fig 4) with maxima occurring in the 1st and 4th quartiles, we compared the maxima of the level scores of the two algorithms in these two quartiles.

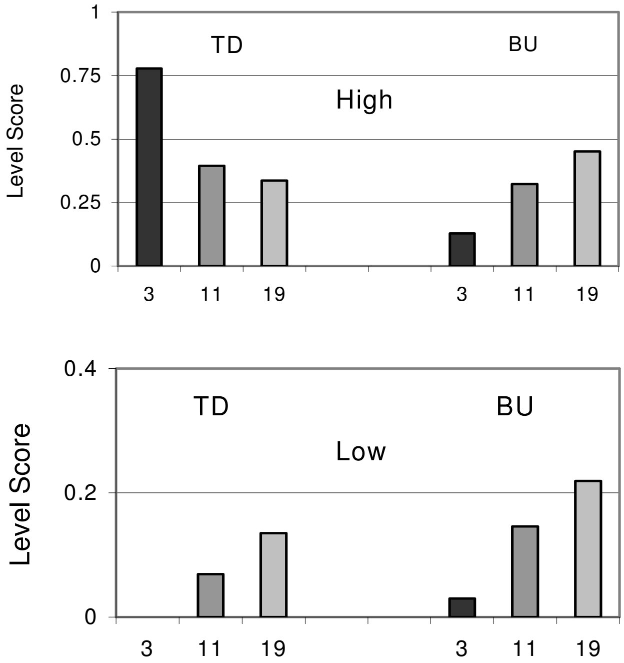


Fig 6. Comparing the two extreme parts of the level scores for the TDQC (top), and BU-A (bottom) algorithms for different levels of granularity (3, 11 and 19 classes). 'High' and 'Low' refer to the 1st and 4th quartiles for the levels in the resulting trees. Note the different scale for the Y-axes.

As depicted in Fig 6, the results show that in the high levels of the tree, the TD algorithm outperforms the BU. The performance of the TD algorithm declines when the granularity from 3 to 19 is increased, whereas the BU performance only improves gradually. At the other end of the scale (low levels of the trees), the scores of both methods improve when granularity increases. However, at all granularity levels, the BU algorithm outperforms TD. Note that in both methods, the overall performance is rather poor for the 4th quartiles of the trees (level score < 0.22). For the 1st quartile, the score of the TDQC reaches 0.78. Similar conclusions were obtained when applying different scoring methods, such as counting the significant nodes in each level (not shown).

Since the high levels reflect a global view of the data whereas low levels account for local aspects, TD algorithms appear to be more appropriate in describing the high level patterns, whereas the opposite holds for local patterns of the data.

Glocal clustering improves the quality of BU algorithms

In the Ion Channel dataset, the instances (proteins) are represented by their distance from each other (E-value). Following the standard BU approach involves jointing sub-clusters solely according to their mutual distance. As suggested above (see Methods), we argue that considering the distances of all sub-clusters in the clusters-merging process may improve the clustering quality. We therefore applied the glocal protocol to the dataset and compared its results with the standard BU algorithms. Fig. 7 displays the trees as generated by the BU (A) and Glocal (B) algorithms. In this case both methods use Euclidian distance and single linkage; similar trend was observed in other combinations.

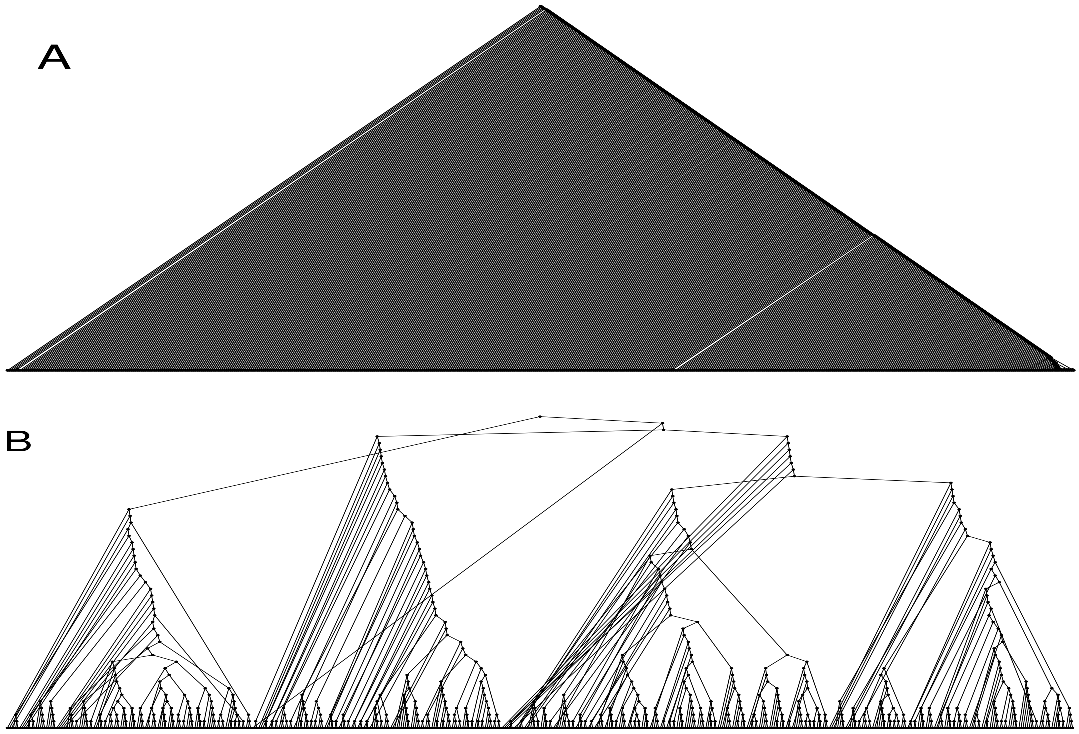


Fig. 7. The hierarchical tree of the BU (A) and glocal (B) algorithms, as applied to the Ion channel dataset. Single linkage wad used in both algorithms.

As Fig 7 shows, the glocal tree is more balanced than the BU tree. Moreover, three clusters are well observed in the glocal tree, while no apparent partition is detected in the BU tree. As the two trees display significantly different structures, we turned to evaluate how well they capture the expert classification at the three resolution levels. Fig .8 displays the tree scores of both algorithms, given the 3 granulation levels.

As displayed in Fig. 8, the glocal protocol improved the clustering results at all granulation levels. The tree, as generated in this way is also more balanced and more informative. Overall, in many other datasets (not shown), we found that adopting this very simple approach may significantly improve clustering results, when comparing to the standard BU implementation.

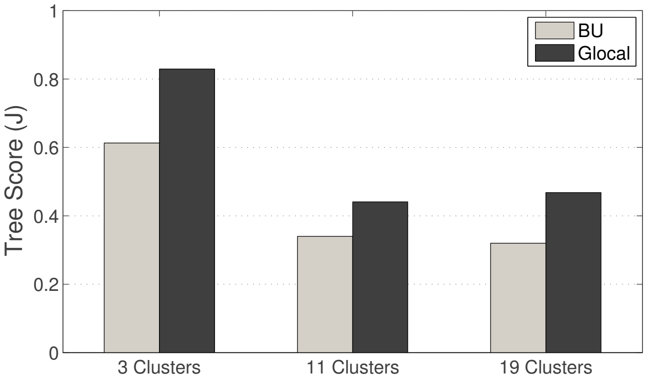


Fig. 8. Tree scores of BU and glocal algorithms for different levels of granularity (3, 11 and 19 classes). Shown are the best results for each approach (single, average or complete).

Biological Interpretations Based on TDQC

With the rapid expansion of available biological data, the reference to an 'expert' often means there has been a combination of automatic and manual efforts. The automatic TDQC algorithm was very successful (score of 0.808) in classifying the coarse granularity of the 518 proteins into 3 classes (Table 2). Nevertheless, the algorithm can also reveal partitions of the data overlooked by these experts (Fig 9). It can be seen in the graph that a group of 35 proteins marked as 'others' is embedded within the sub-tree of 'voltage gated channels' (blue nodes within a brown sub-tree). Inspecting this set of 35 proteins indicates that they are composed of 2 functionally different glutamate ionotropic receptors belonging to NMDA (19 proteins) and Kainate (12 proteins) families (known as NR1-2 and GluR5-7, respectively). For an additional 4 proteins in this set, no clear assignment is provided. Interestingly, an additional set of ionotrophic glutamate receptors set known as AMPA (with 12 proteins, GluR1-4) are separated from the NMDA-Kainate group. Thus, the TDQC partitioned the AMPA ionotrophic glutamate receptors separately from the Kainate and NMDA. Other properties of these receptors including their selectivity, multimeric structures and evolutionary relatedness indeed favor the partition of the AMPA receptors away from the Kinate-NMDA (Zorumski and Thio, 1992). In high quality annotation systems (such as Pfam, SMART and the InterPro integration system) no such separation appeared.

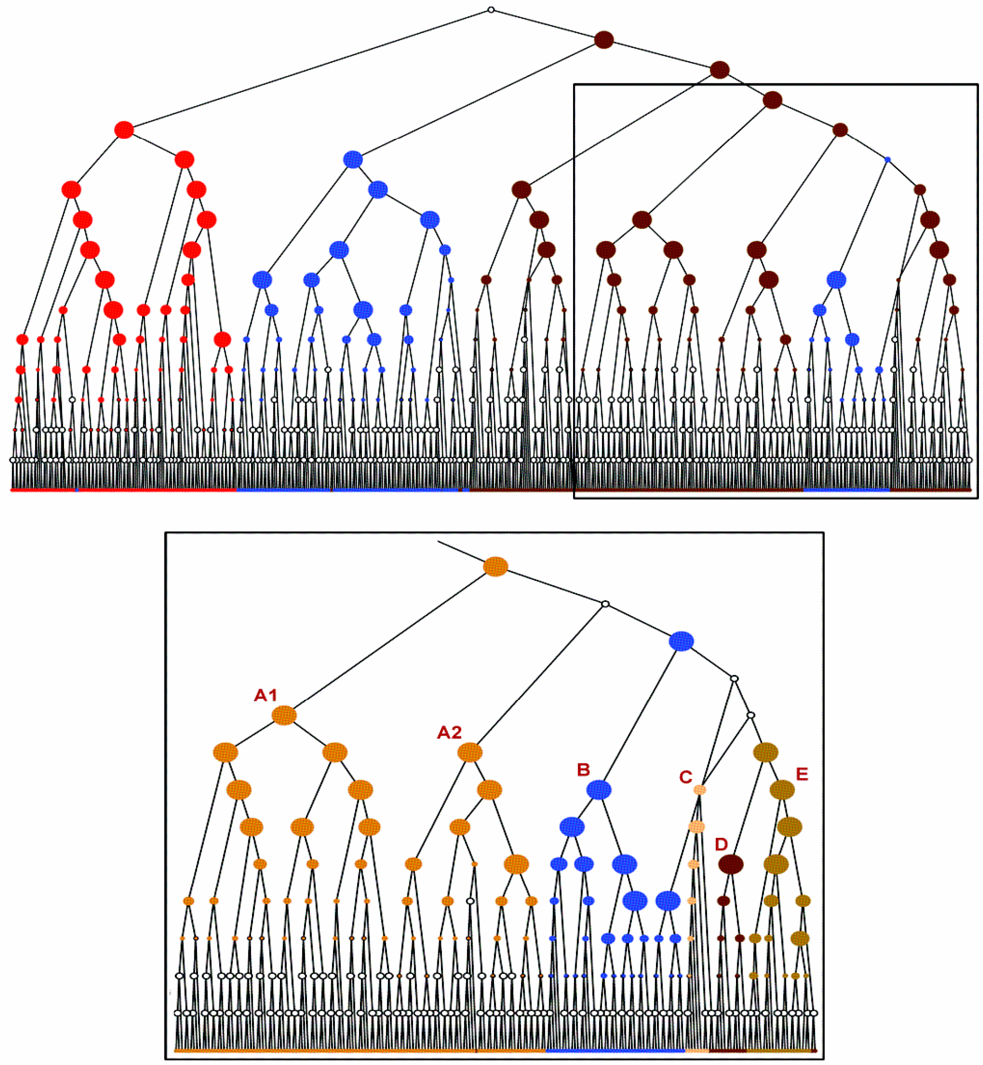


Fig. 9. Hierarchical tree produced by the TDQC algorithm for 518 proteins of ion channels. Red, blue and brown are assigned to the 3 classes: "others", "ligand-gated" and "voltage-gated", respectively. The bottom inset is a zoom of a subset of the tree marked by the frame and according to level of granularity of 11 classes. Sub-trees are all indicated in brown and marked by their identity. A1, A2 - K+ channels ; B – NMDA and Kianate receptors (35 proteins); C - Ryanodine receptors (10 – proteins); D - Na+/H+ exchangers (11 proteins); E - TRP channels (18 proteins). A1 and A2 are separate branches with A1 (73 proteins) including all Kv channels, and A2 with the Cyclic nucleotide-gated channel (51 proteins). Recall, that the top and bottom panels show the same tree.

We further investigated the relationship between the various subtypes of voltage gated channels (marked in brown, Fig. 9) by using a finer granularity of 11 classes (Table 1, supplement). A clear partition was generated by the TDQC and the Kainate-NMDA set

(Fig 9, bottom, marked B). This set is more closely related to the C and D clusters than to A1 and A2. All proteins in cluster A2 are voltage-gated K^+ channels that belong to the Kv1 superfamily and the cyclic regulated channels (whereas the proteins in A1 are Kv1-Kv11). The C cluster comprises a group of 18 TRP channels. All TRP channels are permeable to cations. Although only 2 of the channels (TRPM4 and TRPM5) are impermeable to Ca^{2+}, 2 others (TRPV5 and TRPV6) are highly Ca^{2+} permeable (Owsianik, et al., 2006). Cluster D includes Ryanodine and Inositol 1, 4, 5-trisphosphate (IP3) receptors that are intracellular Ca^{2+} release channels (Berridge, 2004). Cluster E represents a class of Na^+/H^+ exchangers (Orlowski and Grinstein, 2004). Thus the close relationship of the NMDA-Kainate group to Ca^{2+} channels (in clusters C and D) supports their functional relevance and the shared mode of their regulation. Thus, TDQC provides a tree- like structure that not only captures the expert partition but exposes additional connectivity that was overlooked. This group of channels is of special interest as they are targets for pharmaceutical strategies in neurodegenerative diseases and mental pathologies. Their functional partition is far richer than that reflected by their ion conductance properties (Kaczmarek, 2006).

Discussion

We carried out a comparative analysis of five hierarchical clustering algorithms and two nonhierarchical ones, applying them to different types of datasets from various sources. We showed that TD algorithms are consistently superior to BU and nonhierarchical algorithms. In particular, TDQC was found to outperform both TD and BU state-of-the-art algorithms. This applies to data from gene expression, protein families and the stock market.

BU algorithms have some advantages in identifying local relations in the data whereas TD methods capture global patterns. When general patterns are sought, as is the often the case in preliminary stages of data analysis, conventional BU clustering methods should be avoided and replaced by TDs. The latter result in more balanced trees and may be halted – if desired – well before generating the entire tree.

When the data are provided as similarities or distances between instances, we find that a simple manipulation based on all relationships within the data (all distances), may significantly improve the clustering results of the BU approach. This glocal algorithm imposes some global information on the BU making it more competitive with TD algorithms. In summary, global approaches in the exploratory process of clustering, in particular TD or glocal algorithms, are strategies that should not be overlooked.

Although there are ongoing efforts to establish expert hierarchies in various domains, these attempts are riddled with difficulties. High level annotations, often manually catalogued (e.g., GO, UniProt keywords in proteomics) are strongly biased by current knowledge. As a result, that part of the data (in, e.g., protein families) that has been thoroughly studied may possess a rich tree-structure whereas the rest is poorly mapped and weakly annotated. Applying unsupervised methods, such as the TD clustering methods presented here, can leverage the quality of these manually-created mappings. As demonstrated, it can also provide insights into areas that have been missed and correct erroneous annotations.

ClustTree, a graphical Matlab toolbox for applying various hierarchical clustering algorithms and testing their quality is provided and freely available at http://adios.tau.ac.il/clustree/ or http://www.protonet.cs.huji.ac.il/clustree (alternative).

Acknowledgments

We thank Alon Kaufman, Assaf Gottlieb, Nati Linial and the ProtoNet team for stimulating discussions, suggestions, and theoretical assistance and Liat Segal for graphical assistance. R.V. is supported by SCCB, the Sudarsky Center for Computational Biology at the Hebrew University of Jerusalem. This study was supported by the EU Framework VI consortium.

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.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res*, **25**, 3389-3402.

Apweiler, R., Biswas, M., Fleischmann, W., Kanapin, A., Karavidopoulou, Y., Kersey, P., Kriventseva, E.V., Mittard, V., Mulder, N., Phan, I. and Zdobnov, E. (2001) Proteome Analysis Database: online application of InterPro and CluSTr for the functional classification of proteins in whole genomes, *Nucleic Acids Res*, **29**, 44-48.

Benjamini, Y. and Hochberg, Y. (1995) Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, *Journal of the Royal Statistical Society. Series B (Methodological)*, **57**, 289-300.

Berridge, M. (2004) Conformational Coupling: A Physiological Calcium Entry Mechanism, *Sci. STKE*, **2004**, pe33-.

Boley, D. (1998) Principal Direction Divisive Partitioning. *Data Mining and Knowledge Discovery*. 325 - 344.

Chipman, H. and Tibshirani, R. (2006) Hybrid hierarchical clustering with applications to microarray data, *Biostat*, **7**, 286-301.

Cimiano, P., Hotho , A. and Staab, S. (2004) Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text. *Proceedings of the European Conference on Artificial Intelligence*. 435--439.

D'Haeseleer, P. (2005) How does gene expression clustering work?, *Nat Biotechnol*, **23**, 1499-1501.

Duda, R.O., Hart, P.E. and Stork, D.G. (2000) *Pattern Classification*. Wiley-Interscience.

Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns, *PNAS*, **95**, 14863-14868.

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.286.5439.531, *Science*, **286**, 531-537.

Handl, J., Knowles, J. and Kell, D.B. (2005) Computational cluster validation in post-genomic data analysis, *Bioinformatics*, **21**, 3201-3212.

Hansen, P. and Delattre, M. (1978) Complete-link cluster analysis by graph coloring, *J. American Stat. Ass.*, **73**, 397 - 403.

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**.

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

Kaczmarek, L.K. (2006) Non-conducting functions of voltage-gated ion channels, **7**, 761-771.

Kaplan, N., Friedlich, M., Fromer, M. and Linial, M. (2004) A functional hierarchical organization of the protein sequence space, *BMC Bioinformatics*, **5**, 196.

Kruskal, J.B. and Wish, M. (1981) *Multidimensional scaling*. Sage Publications, Beverly Hills.

Landauer, T.K., Foltz P. W. and Laham, D. (1998) Introduction to Latent Semantic Analysis, *Discourse Processes*, **25**, 259-284.

MathWorld (2007) Matlab Statistics Toolbox. MathWorld.

Orlowski, J. and Grinstein, S. (2004) Diversity of the mammalian sodium/proton exchanger SLC9 gene family, *PflÃ¼gers Archiv European Journal of Physiology*, V447, 549-565.

Owsianik, G., Talavera, K., Voets, T. and Nilius, B. (2006) PERMEATION AND SELECTIVITY OF TRP CHANNELS, *Annual Review of Physiology*, **68**, 685-717.

Planet, P.J., DeSalle, R., Siddall, M., Bael, T., Sarkar, I.N. and Stanley, S.E. (2001) Systematic Analysis of DNA Microarray Data: Ordering and Interpreting Patterns of Gene Expression, *Genome Res.*, **11**, 1149-1155.

Ren, Q. and Paulsen, I.T. (2005) Comparative Analyses of Fundamental Differences in Membrane Transport Capabilities in Prokaryotes and Eukaryotes, *PLoS Computational Biology*, **1**, e27.

Rune, M. (2007) Methods, algorithms and tools in computational proteomics: A practical point of view, *PROTEOMICS*, **7**, 2815-2832.

Sasson, O., Vaaknin, A., Fleischer, H., Portugaly, E., Bilu, Y., Linial, N. and Linial, M. (2003) ProtoNet: hierarchical classification of the protein space, *Nucleic Acids Res*, **31**, 348-352.

Savaresi, M.S. and Boley, D. (2004) A comparative analysis on the bisecting K-means and the PDDP clustering algorithms., *Intelligent Data Analysis*, **8**, 345-362.

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

Slonim, N., Atwal, G.S., Tkacik, G. and Bialek, W. (2005) Information-based clustering, *PNAS*, **102**, 18297-18302.

Steinbach, M., Karypis, G. and Kumar, V. (2000) A comparison of document clustering techniques. *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Boston.

Torrente, A., Kapushesky, M. and Brazma, A. (2005) A new algorithm for comparing and visualizing relationships between hierarchical and flat gene expression data clusterings, *Bioinformatics*, **21**, 3993-3999.

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

Zhao, Y. and Karypis, G. (2002) Evaluation of hierarchical clustering algorithms for document datasets. In, *Proceedings of the eleventh international conference on Information and knowledge management* ACM Press, McLean, Virginia, USA 515-524

Zorumski, C.F. and Thio, L.L. (1992) Properties of vertebrate glutamate receptors: Calcium mobilization and desensitization, *Progress in Neurobiology*, **39**, 295-336.

Chapter 4

Clustering Evaluation

This chapter contains the following research papers:

[4A] **Roy Varshavsky**, Michal Linial and David Horn. "*Clustering Algorithms Optimizer: A Framework for Large Datasets*" (2007, ISBRA, Lecture Notes in Bioinformatics (4463), 85-96).

[4B] **Roy Varshavsky**, Michal Linial, David Horn. "*COMPACT: A Comparative Package for Clustering Assessment*" (2005, ISPA, Lecture Notes in Computer Science (3759), 159-167).

Section 4.1

Clustering Algorithms Optimizer: A Framework for Large Datasets

Clustering Algorithms Optimizer: A Framework for Large Datasets

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

[1] School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel
royke@cs.huji.ac.il
[2] School of Physics and Astronomy, Tel Aviv University, Israel
[3] Deptartment of Biological Chemistry, Institute of Life Sciences, The Hebrew University of Jerusalem, Israel

Abstract. Clustering algorithms are employed in many bioinformatics tasks, including categorization of protein sequences and analysis of gene-expression data. Although these algorithms are routinely applied, many of them suffer from the following limitations: (i) relying on predetermined parameters tuning, such as a-priori knowledge regarding the number of clusters; (ii) involving nondeterministic procedures that yield inconsistent outcomes. Thus, a framework that addresses these shortcomings is desirable. We provide a data-driven framework that includes two interrelated steps. The first one is SVD-based dimension reduction and the second is an automated tuning of the algorithm's parameter(s). The dimension reduction step is efficiently adjusted for very large datasets. The optimal parameter setting is identified according to the internal evaluation criterion known as Bayesian Information Criterion (BIC). This framework can incorporate most clustering algorithms and improve their performance. In this study we illustrate the effectiveness of this platform by incorporating the standard K-Means and the Quantum Clustering algorithms. The implementations are applied to several gene-expression benchmarks with significant success.

Abbreviations and Keywords: Bayesian Information Criterion (BIC), Quantum Clustering (QC), Optimal K-Means (OKM), Optimal Quantum Clustering (OQC), Principal Component Analysis (PCA), Singular Value Decomposition (SVD).

1 Introduction[1]

In the field of genomics and proteomics, as well as in many other disciplines, categorization is a fundamental challenge. Categorization is defined as systematically arranging elements (data-points) into specific groups. Clustering, being an unsupervised learning problem, may be regarded as a special case of categorization with unknown

[*] Corresponding author.

[1] **Availability and Supplementary material:** The framework has been implemented in MATLAB (Version 6.5), and is freely available at http://adios.tau.ac.il/compact/framework

I. Măndoiu and A. Zelikovsky (Eds.): ISBRA 2007, LNBI 4463, pp. 85–96, 2007.

labels (for further details see [1, 2]). Some algorithms such as CLICK [2], CTWC [3, 4] and CAST [5] were primarily developed for large sets of biological data while others were adopted from other fields (e.g., K-Means, Fuzzy C-means [6], Agglomerative Hierarchical Clustering, Self Organized Maps). One of the algorithms that we will expand on is Quantum Clustering (QC), the effectiveness of which has been demonstrated on gene-expression data [7, 8].

In large scale gene-expression tasks, clustering algorithms are useful for diagnosis of different samples (e.g., differentiating sick and healthy tissues, associating tissues with subtypes of a disease) as well as revealing functional classes of genes among the thousands often used in experimental settings [9].

Methods for collecting expression levels on a genome-wide level have been rapidly improving, leading to increased amounts of data to be analyzed. Additionally, much of the biological data is represented in high dimensions. Some clustering algorithms do not perform well when applied to large high-dimensional datasets. In particular, several model-based algorithms that are shown to be very efficient on limited size datasets [10], are found unfeasible when large scale datasets arc introduced (for computational complexity discussion see [11] and supplementary). The hope is that efficient preprocessing will address the task of computational feasibility while efficiently remove noise, thus allowing exposure of meaningful features of the data.

It would be presumptuous to propose one preprocessing protocol that works for all kinds of data. Different preprocessing methods are based on averaging and variance standardization, excluding genes with low variance between conditions [2], PCA, Fourier transforms [12], and more.

One fundamental preprocessing direction is dimension reduction. Ding *et al.* claim that the dimension should be correlated with the expected number of clusters [13]. However, this may not hold for real biological data, since this argument is based on a model in which data are generated by independent Gaussian distributions. Moreover, in many cases the number of clusters is unknown.

Several efforts to develop efficient and accurate filtering schemes and compression tools have been proposed [14, 15]. A routine scheme for gene-expression data (including commercial analysis tools provided by various platforms) is to filter elements in a supervised manner. For example, genes whose variance is below a certain threshold for different experimental conditions are discarded. Obviously, such filtering is often biased and misses a genuine property of the data.

In addition to preprocessing, clustering algorithms usually require selecting a set of parameters, thus turning each application into a set of subjective choices. If no prior knowledge is available, assessing the correct number of clusters (e.g., as required by the K-Means algorithm), is almost impossible. This choice is avoided by hierarchical algorithms that propose some O(N) possible partitions[2] of varying sizes, and the decision on the best partition is user determined.

Several of the most successful algorithms in the field of gene-expression do not explicitly accept the number of clusters K as an input; however this number is directly derived from their parameters. Amongst them are *(i)* the CAST algorithm [5], in which the affinity threshold determines the number of clusters, *(ii)* the CLICK

[2] In the paper N refers to the number of elements in the data, and K denotes the number of clusters.

algorithm [2], in which the homogeneity value determines K by controlling the kernels and the definition of singletons. *(iii)* The CTWC algorithm [4] where some parameters (such as stability threshold and minimal group size) determine K, and *(iv)* QC [7] where the Parzen window size (σ) determines the number of clusters.

Moreover, algorithms such as K-Means, Fuzzy C-Means and others, being nondeterministic, are inconsistent as they depend on starting points and other stochastic factors. Some methods such as averaging clustering results, following a majority rule, or applying other heuristics [16] have been suggested.

Since different results may be obtained by the numerous clustering algorithms that exist, evaluation of this variety is an essential step of the analysis [17, 18], and a reliable method is required. In this study we present a framework to overcome the pitfalls described above by (i) a generic method for preprocessing and (ii) a measure based on an internal criterion that can be incorporated in any clustering algorithm.

2 Methods

Our proposed framework includes two interrelated steps: preprocessing and parameter tuning. We outline the rationale of the method and describe its implementation on two different kinds of clustering algorithms.

2.1 Preprocessing

Singular Value Decomposition (SVD) serves as a good and efficient preprocessing step and is useful for dimension reduction [8, 12, 19].

SVD represents any real matrix X as a product $X=U\Sigma V^T$, where U and V are orthonormal matrices and Σ is a diagonal matrix whose eigenvalues s_i (singular values) appear in decreasing order. The columns of U and V define two independent vector spaces. This decomposition is unique (up to overall phases) and holds for any real matrix of size m by n. The number of non-zero entries in Σ equals the rank of X. A common application of SVD is dimension reduction: this is performed by replacing Σ with a truncated version where only a small number (r) of leading singular values is retained and the rest are replaced by zeros. The resulting reconstructed matrix X' ($X'=U\Sigma' V^T$), is the best least-mean-squares approximation of X obtainable by any matrix of rank r.

We focus our attention on the matrices U and V. In a problem where X is a matrix of m genes by n samples, U and V form representations of gene and sample spaces respectively. It is within these spaces, now reduced to rank r that we look for cluster structures [8].

How does one choose the rank r of the truncated space? The singular values s_i have the meaning of standard deviations. Defining the relative variance V_i of component i (see Fig 1A and supplementary), one may come up with several principles for truncation.

$$V_i = s_i^2 / \sum_{j=1}^{N} s_j^2 \tag{1}$$

Wall [12] suggested the following guidelines: (1) ignore components beyond the point where the cumulative relative variance becomes larger than a certain threshold (e.g. 85%), (2) ignore components with relative variance below a certain threshold (e.g. 1%), or (3) stop when a sudden decrease is observed in the relative variance graph. We suggest using SVD- entropy [19] as a guide for choosing among the possibilities.

$$E(Data) = -\frac{1}{\log(N)}\sum_{i=1}^{N} V_i \log(V_i) \tag{2}$$

E 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. We find that in gene-expression datasets, entropy values are higher than 0.5, reflecting a disordered distribution. If *E* is very low, a sudden decrease in the spectrum is a good indicator for the best *r* values. Otherwise we prefer criteria *(1)* and *(2)*.

Truncation to dimension *r* is equivalent to projecting the vectors of our problem (e.g. the genes or samples vectors) onto an r-dimensional subspace. The vectors, as defined in this subspace, have different norms. It is preferable to renormalize the vectors, i.e. project them onto the unit hyper-sphere in r-space. This approach considers similarity between vectors in the truncated space in terms of the cosine of the angle between them, and is consistent with the standard application of Latent Semantic Analysis (LSA) [20]. It is worth mentioning that, although we suggest using SVD, other truncation methods may be used (e.g., Fourier transforms, PCA).

2.2 Parameter Tuning

The validity and reliability of clustering algorithms may be questioned on two grounds: *(1)* subjectivity, i.e. using supervised criteria in the parameter setting and *(2)* inconsistency, i.e. obtaining different results upon repeated application of nondeterministic algorithms.

In order to reduce these pitfalls to a minimum, we suggest using an internal criterion. The criterion we choose to adopt is the Bayesian Information Criterion (BIC). Fraley and Raftery [21] developed it in a model-based analysis that assumed the data to be generated by a mixture of underlying normal probability distributions. The parameters of the underlying distributions were set by an EM algorithm. The BIC criterion is used to evaluate the number of clusters and the quality of the suggested clustering. BIC is defined as follows:

$$BIC \equiv 2l_M(x,\hat{\Theta}) - m_M \log(N) \approx 2\log p(x \mid M) + const \tag{3}$$

where $l_M(x,\Theta)$ is the mixture log likelihood (of the data *x* and the predicted model Θ), which is maximized under the constraint that m_M (a function of the number of independent parameters[3]), is minimized. It is assumed that a higher BIC score reflects better clustering quality. Recently, Teschendorff *et al.* have applied an EM algorithm to find a partition that maximizes the BIC criterion [10]. Here we do not optimize the

[3] We choose m_M=*dim*K* (K+dim),* where dim is the number of dimensions and K is the number of clusters.

BIC score. Trusting the clustering algorithms we just use this score, in a way befitting the algorithms, to find the best clustering parameters.

3 Implementation

We demonstrate our method on two fundamentally different clustering algorithms. They differ in some fundamental aspects thus testing the generality of our framework.

Optimized K-Means (OKM)
K-Means is a very popular, fast and intuitive algorithm. This naïve algorithm has two known drawbacks: First, it requires the number of clusters as an input, and thus is limited to scenarios where external knowledge is available. Secondly, the algorithm is nondeterministic, and is thus inconsistent.

The OKM implementation applies the K-Means algorithm 50 times for each number of clusters (K=1 to 20 in our examples) and computes the BIC score for each application. The application that leads to the maximal BIC score is considered to be the optimal solution.

Optimized QC (OQC)
The QC algorithm [7] uses the Schrödinger equation to provide an effective clustering description of the data. It requires one parameter, σ, a Parzen window width. This parameter controls the number of clusters that are identified by the algorithm with larger values of σ yielding fewer clusters. Different σ may also yield the same number of clusters but different clustering assignments (see Fig. 2B). Contrary to K-Means this algorithm is deterministic, has less constraints than K-means (since noise is integrated within the model), and does not assume spherical properties of the clusters. Recently, a variation of the algorithm's convergence, using the mean-shift approach, was suggested [22]. Here we employ the standard implementation [7].

OQC consists of applying QC once for a set of σ values (50 values in the range of 0.1 to 0.9, in our examples), and computes the BIC score for each σ. The maximal BIC is considered as the optimal solution.

4 Results

Here we describe our results on three gene-expression datasets that are well known benchmarks. In the first [23] and the second [24] examples, samples were clustered (2 and 4 clusters, respectively) while in the third dataset [25] clustering was performed on the genes. All three cases have assignments that were manually curated. The assignments serve to estimate the performance of the clustering algorithms, using the Jaccard score which reflects the 'intersection over union' between the algorithm's clustering assignments and the expected classification[4]:

$$Jaccard = \frac{n_{11}}{n_{11} + n_{01} + n_{10}} \tag{4}$$

[4] We refer to supplementary material for further explanation.

4.1 The Colon Dataset of Alon et al. (1999)

In the dataset of [23], 62 gene-expression samples were taken from colon cancer patients. 40 of them were taken from sick tissues, and 22 from healthy tissues. Each sample contains the expression of 7479 genes. We follow [23, 24] who chose 2000 genes with the highest confidence in the measured expression levels.

In order to emphasize the influence of preprocessing on the clustering results, we compare SVD (see methods) with Principal Components Analysis (PCA)[5]. Fig 1A displays the singular values of the [2000x62] matrix.

The compression guidelines (see methods), suggests that only 2 or 3 components may be needed for a good description of the data (the relatively low entropy: 0.28, see equation 2). This yields compression rates of $1x10^{-3}$ and $1.5x10^{-3}$, respectively.

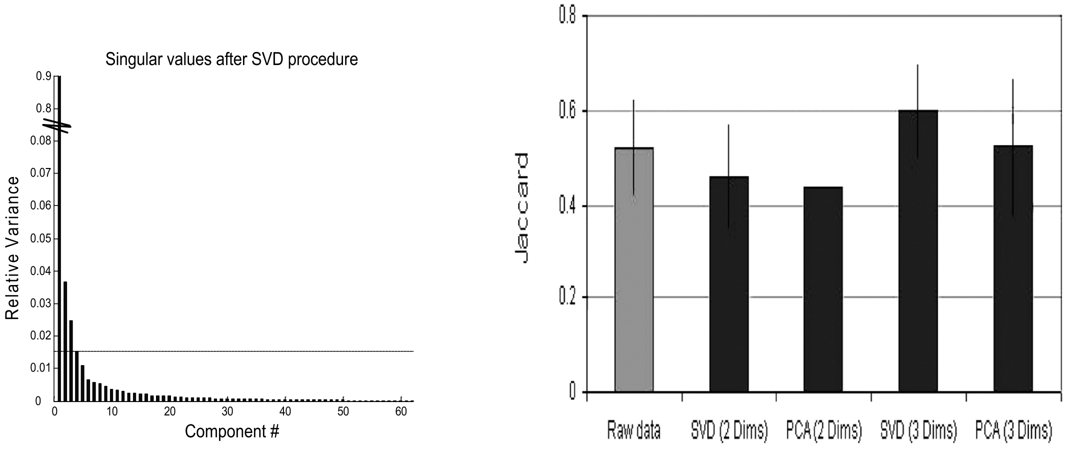


Fig. 1. A. (left) Singular values of the colon dataset (dashed line denotes the 'cut' decision). **B.** (right) Jaccard scores of the KM on raw data (left bar) and different preprocessing options.

As shown in Fig. 1A, preprocessing procedure influences the clustering quality. We conclude that this step deserves substantial attention. Moreover, when selecting the correct compression method (SVD in 3 dimensions), the clustering results are improved, as reflected by the increase in the Jaccard score (from 0.52 to 0.6).

The optimal results are obtained for SVD reduction to 3 dimensions. At this stage, the data are compressed to 62 vectors on a 3 dimensional unit sphere. Fig. 2A displays the OKM results (50 executions for 2-20 putative clusters) for different choices of K. For each K the maximal BIC of all 50 trials was chosen. The overall maximal BIC value is obtained for K=2. Note that the farther the number of clusters is from the correct solution, the larger is the dispersion of the corresponding BIC values. Comparing the internal (BIC) and external (Jaccard) criteria, one finds that the K=2 assignments were also the closest to the experts opinion. This testifies to the usefulness of BIC as an indicator of the proper clustering of the data.

[5] Matlab code: `princomp(zscore(X'X))`.

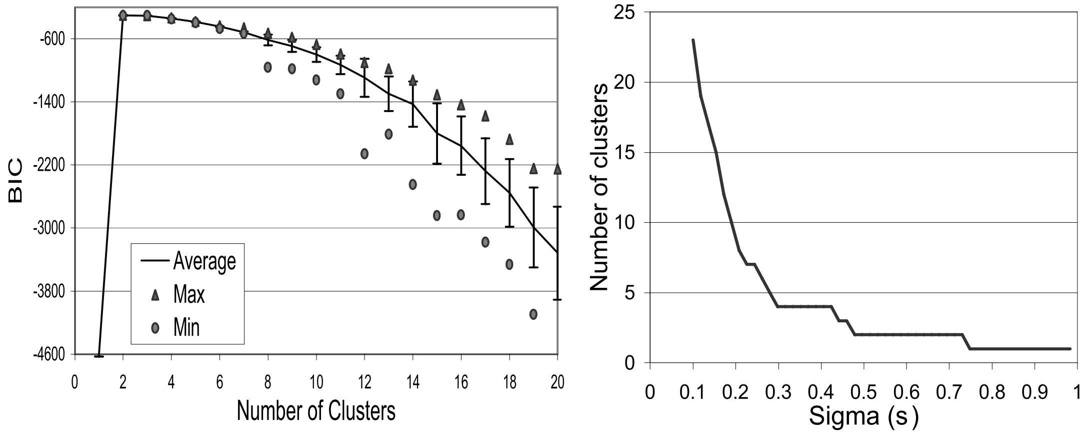


Fig. 2. A. (left) BIC Values when applying OKM (SVD reduced to 3 dimensions) on the colon dataset. **B.** (right) The number of clusters obtained in the colon dataset as a function of the σ input parameter of the QC algorithm.

Next we apply OQC to the compressed colon dataset. Recall that QC is a deterministic algorithm, thus, a single application is required for each σ value. Fig. 2B displays the number of clusters when varying σ. Note that different σ values may lead to the same number of clusters but different assignments, hence BIC may vary when the number of clusters remains constant.

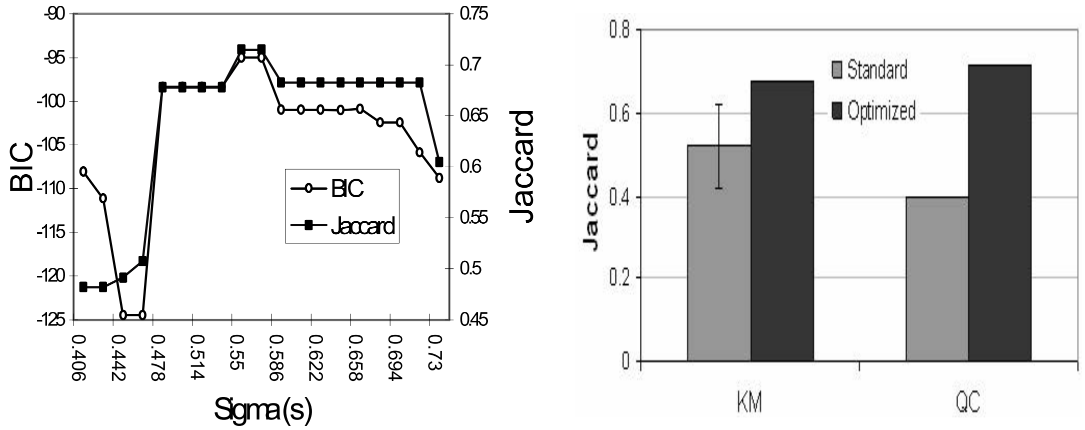


Fig. 3. A. (left) Comparison of the internal (BIC) and external (Jaccard) criteria for the colon dataset (OQC). **B.** (right) Comparison of the standard and optimized versions of the KM and QC algorithms.

Both BIC and Jaccard scores display the same behavior in the neighborhood of their maximal values (Fig. 3A). The maximal BIC was obtained for σ=0.55, where QC leads to 2 clusters. The corresponding Jaccard score for this σ is 0.715.

Since both OKM and OQC share the same preprocessing step, their clustering results can be compared. The maximal BIC value achieved by OQC is higher than the one achieved by OKM (-95 and -300, respectively). Similarly, the Jaccard score of the

OQC is higher than the one of OKM (0.715 and 0.678, respectively). Fig. 3B compares these results with what the same algorithms obtain on the original datasets without preprocessing (0.52 and 0.4 for KM and QC, respectively). The results are even more impressive when compared to other state-of-the-art algorithms (Table1).

Table 1. Jaccard scores of various algorithms when applied to the Alon dataset

Method	**Jaccard**
K-Means (raw data, 50 repeats)	0.52 (0.1)
OKM (Preprocessing & BIC)	0.678
QC (raw data)	0.4
OQC (Preprocessing & BIC)	**0.715**
CLICK [2]	0.64
CAST [2,5]	0.682
CTWC ([4], and[6])	0.508

4.2 The Leukemia Dataset of Golub et al., 1999

The dataset of Golub *et al.* has served as a benchmark for several clustering methods [2, 4 and 24]. 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 clustering task is to find the four cancer groups within the 72 patients in a [7129x72] gene expression matrix. We select the first five eigenvectors, achieving a compression rate of $7x10^{-4}$ (from [7129x72] to [5x72]).

BIC is maximized for K=2 in OKM, as is the Jaccard score (Fig. 4A). Hence we conclude that OKM can identify only the two major groups in the data and cannot detect a partition into four groups. This finding is consistent with the CAST and CLICK algorithms that have also failed to identify the subtypes [2]

Since QC cannot be applied to the raw dataset, preprocessing is of essence. OQC proves to be very effective. As displayed in Fig. 4B, the correlation between the BIC and the Jaccard scores is quite high around the maximum of both curves. Moreover, the maximum BIC is at σ =0.548, which dictates partitioning into 4 clusters, similar to what would be expected from the data. The corresponding Jaccard score for this σ is 0.69 (Fig. 4B). 4 clusters are predicted by QC throughout the range $0.47<\sigma<0.56$.

4.3 The Yeast Dataset of Spellman et al. (1998)

The dataset of [25] presents a somewhat more challenging task than the previous examples, since we examine our method on clustering of genes. Spellman *et al.* identified 798 genes as cell cycle regulated and assigned them to 5 different stages of the yeast cell cycle (M/G1, G1, S, G2 and M). Expression levels of these genes were recorded at 72 time points, yielding a [798x72] matrix.

[6] http://www.weizmann.ac.il/physics/complex/compphys/ctwc/

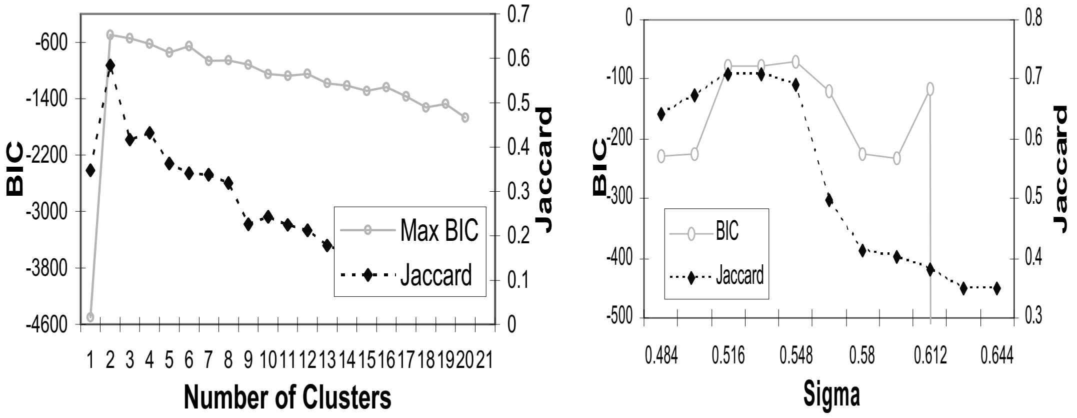


Fig. 4. A. (left) BIC and Jaccard scores of the Golub dataset (OKM), **B.** (right)Comparison of internal (BIC) and external (Jaccard) criteria of the leukemia dataset (OQC)

Contrary to the first examples, the distribution of relative variances is gradual and the entropy is significantly higher (0.705, see supplement). This result is consistent with the argument that high entropy reflects data that were preprocessed, since genes were intentionally selected by their functional annotation. We selected the first four leading eigenvectors (note the dashed line in the figure) achieving a compression rate of $5x10^{-2}$ (from [798x72] to [798x4]).

The external expert [25] suggests that there are 5 groups of cell cycle related genes. When applying the OKM protocol to the compressed dataset a maximized BIC is observed at 6 clusters. Comparing to the standard application of K-Means, the OKM shows no improvement: both applications yield Jaccard scores of 0.4.

Application of OQC to the compressed dataset yields a somewhat different result than that of OKM. BIC is maximized at σ=0.5, where 4 clusters are identified. Taking a closer look at the OQC clusters suggests that the S and G2 stages are joined by QC into one cluster. Here the correlation between the BIC and Jaccard scores is not perfect (see supplementary). Nevertheless, the Jaccard score it yields is relatively high (0.5 comparing to 0.4 in many other algorithms, see supplement table).

5 Conclusion

We present a general 'clustering improver' scheme. This unsupervised, data-driven two-step clustering framework uses intrinsic properties of the dataset to determine the SVD-based compression. After dimension reduction, several iterations of a clustering algorithm are applied, each with a different parameter. They are then compared with each other by the BIC criterion. The parameter that yields the best BIC score is chosen and is declared to be the optimal one. This generic framework is also computationally efficient: it processes these large-scale datasets on a standard PC in less than a minute (e.g., 50 runs of each of the different number of clusters in OKM).

Preprocessing of experimental data is an essential step. The raw data often come in a large-scale, un-normalized and noisy representation. These distractions have to be treated. Nevertheless, due to the diversity of the experiments one cannot provide a universal preprocessing method. In our study, we emphasize the importance of compression, and present some examples of the variations that different preprocessing methods can yield. We recommend SVD-based compression, which provides a normalized, filtered and ultra-compressed representation of the data. We also suggest guidelines regarding the extent of the compression.

The second step of our methodology is parameter tuning, which is based on the BIC score. Choosing this score has two advantages: *(1)* being an internal measurement, it allows an unbiased, automated method with no external intervention, and *(2)* its capability to be computed after the algorithm has terminated its application allows this independent criterion to be ‘plugged in’ to any clustering algorithm.

BIC is useful for finding the best solution amongst many local maxima, for both deterministic and nondeterministic clustering algorithms. Some heuristics are proposed in order to overcome the inconsistency problem of nondeterministic algorithms. In cases where many applications of the same algorithm lead to suboptimal solutions and only a few suggest good solutions, BIC maximization represents considerable improvement over other methods such as majority voting. Even if BIC does not point to the best clustering solution, it chooses one that is close to the best. It can therefore assist in narrowing down the search for best parameters.

Our methodology is especially well adapted to algorithms that assume spherical distribution (e.g., K-Means) of clusters, but it can be applied to algorithms that do not assume such a distribution. Surprisingly, it performs very well for methods that do not subsume spherical clustering such as QC and SOM (not shown). The optimized algorithms described here outperform the published results of CTWC, CLICK and CAST. We assume the same methodology to the latter algorithms could improve their performance even further.

Nevertheless, we identify some limitations. First, as we have not suggested any modification in any clustering algorithm per se, the improvement is bounded to the algorithm’s best performance. If the solution space does not describe the underlying structure of the dataset, we cannot obtain a high quality solution.

Second, the BIC score assumes a specific hyper-elliptic organization of clusters. When, as in the yeast dataset, clusters have different distributions, BIC has less descriptive strength. In such cases BIC may not fit the properties of the dataset. Third, the BIC value, computed by the EM method, usually cannot converge when the number of dimensions surpasses some threshold (of the order of 10). An efficient preprocessing is therefore a prerequisite for the BIC to be computed.

Finally, since BIC fits a model to a specific data distribution, it cannot be used to compare models of different datasets. For the same reasons it cannot be used to choose among different preprocessing methods or truncated dimensions.

Different clustering algorithms are currently included in analysis suites that are applied by experimentalists to gene expression data. A standard practice is to apply several algorithms with a few configurations and choose among them on the basis of some known classification. Our framework may serve as a platform for systematic comparison between different clustering algorithms. In all comparisons, analysis is applied to an identical experimental benchmark. The large variation in performance of each algorithm supports the notion that there is no 'one-size-fits-all' method. This study attempts to reduce the subjectivity in data interpretation by providing a platform for comparisons that can be adopted by any algorithm.

Acknowledgements. We thank Noam Kaplan, Alon Kaufman and Menachem Fromer for critical reading the manuscript. R.V. has been awarded a fellowship by the Sudarsky Center for Computational Biology. This work is partially supported by EU Framework VI, DIAMONDS consortium.

References

1. Jain AK, Dubes RC: Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice Hall; 1988.
2. Sharan R, Shamir R: CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis. In*: 2000*: AAAI Press, Menlo Park, CA; 2000: 307–316.
3. Blatt M, Wiseman S, Domany E: Superparamagnetic Clustering of Data. *Physical Review Letters* 1996, 76:3251–3254.
4. Getz G, Levine E, Domany E: Coupled two-way clustering analysis of gene microarray data. *PNAS* 2000, 97(22):12079-12084.
5. Ben-Dor A, Shamir R, Yakhini Z: Clustering Gene Expression Patterns. *Journal of Computational Biology* 1999, 6(3-4):281-297.
6. Dembele D, Kastner P: Fuzzy C-means method for clustering microarray data. *Bioinformatics* 2003, 19(8):973-980.
7. Horn D, Gottlieb A: Algorithm for data clustering in pattern recognition problems based on quantum mechanics. *Physical Review Letters* 2002, 88(1).
8. Horn D, Axel I: Novel clustering algorithm for microarray expression data in a truncated SVD space. *Bioinformatics* 2003, 19(9):1110-1115.
9. Eisen MB, Spellman PT, Brown PO, Botstein D: Cluster analysis and display of genome-wide expression patterns. *PNAS* 1998, 95(25):14863-14868.
10. Teschendorff AE, Wang Y, Barbosa-Morais NL, Brenton JD, Caldas C: A variational Bayesian mixture modelling framework for cluster analysis of gene-expression data. *Bioinformatics* 2005, 21(13):3025-3033.
11. Zhong S, Ghosh J: A unified framework for model-based clustering. *Journal of Machine Learning Research* 2003, 4(964287):1001-1037.
12. Wall M, Rechtsteiner A, Rocha L: Singular Value Decomposition and Principal Component Analysis. In: *A Practical Approach to Microarray Data Analysis.* Edited by Berrar D, Dubitzky W, Granzow M: Kluwer; 2003: 91-109.
13. Ding C, He X, Zha H, Simon H: Adaptive dimension reduction for clustering high dimensional data. In: *IEEE International Conference on Data Mining: 2002*; 2002: 107-114.
14. Xing EP, Karp RM: CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics* 2001, 17(90001):S306-315.
15. Plagianakos VP, Tasoulis DK, M.N. V: Hybrid dimension reduction approach for gene expression data classification. In: *International Joint Conference on Neural Networks 2005, Post-Conference Workshop on Computational Intelligence Approaches for the Analysis of Bioinformatics: 2005.*
16. Zhong W, Altun G, Harrison R, Tai PC, Pan Y: Improved K-means Clustering Algorithm for Exploring Local Protein Sequence Motifs Representing Common Structural Property. In: *IEEE Transactions on NanoBioscience: 2005*; 2005: 255-265.
17. Handl J, Knowles J, Kell DB: Computational cluster validation in post-genomic data analysis. *Bioinformatics* 2005, 21(15):3201-3212.

18. Varshavsky R, Linial M, Horn D: COMPACT: A Comparative Package for Clustering Assessment. In: *Lecture Notes in Computer Science.* 3759 ed: Springer-Verlag; 2005: 159-167.
19. Alter O, Brown PO, Botstein D: Singular value decomposition for genome-wide expression data processing and modeling. *PNAS* 2000, 97(18):10101-10106.
20. Landauer TK, Foltz P. W., Laham D: Introduction to Latent Semantic Analysis. *Discourse Processes* 1998, 25:259-284.
21. Fraley C, Raftery AE: How many clusters? Which clustering method? - Answers via Model-Based Cluster Analysis. In: *Computer Journal.* vol. 41; 1998: 578-588.
22. Barash, D. and D. Comaniciu. *Meanshift clustering for DNA microarray analysis*. In Computational Systems Bioinformatics Conference (CSB) 2004: IEEE.
23. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS* 1999, 96(12):6745-6750.
24. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA *et al*: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science* 1999, 286(5439):531-537.
25. Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B: Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. *Mol Biol Cell* 1998, 9(12):3273-3297.

Section 4.2

COMPACT: A Comparative Package for Clustering Assessment

COMPACT: A Comparative Package for Clustering Assessment

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

[1] School of Computer Science and Engineering,
The Hebrew University of Jerusalem 91904, Israel
royke@cs.huji.ac.il
[2] Dept of Biological Chemistry, Institute of Life Sciences,
The Hebrew University of Jerusalem 91904, Israel
michall@cc.huji.ac.il
[3] School of Physics and Astronomy, Tel Aviv University, Israel
horn@post.tau.ac.il

Abstract. There exist numerous algorithms that cluster data-points from large-scale genomic experiments such as sequencing, gene-expression and proteomics. Such algorithms may employ distinct principles, and lead to different performance and results. The appropriate choice of a clustering method is a significant and often overlooked aspect in extracting information from large-scale datasets. Evidently, such choice may significantly influence the biological interpretation of the data. We present an easy-to-use and intuitive tool that compares some clustering methods within the same framework. The interface is named COMPACT for **Com**parative-**Pa**ckage-for-**C**lustering-Assessmen**t**. COMPACT first reduces the dataset's dimensionality using the Singular Value Decomposition (SVD) method, and only then employs various clustering techniques. Besides its simplicity, and its ability to perform well on high-dimensional data, it provides visualization tools for evaluating the results. COMPACT was tested on a variety of datasets, from classical benchmarks to large-scale gene-expression experiments. COMPACT is configurable and expendable to newly added algorithms.

1 Introduction

In the field of genomics and proteomics, as well as in many other disciplines, classification is a fundamental challenge. Classification is defined as systematically arranging entities (data-points) into specific groups. Clustering, being an unsupervised learning problem, may be regarded as a special case of classification with unknown labels (for more details see [1], [2]). In gene expression microarray technology, a hierarchical clustering algorithm was first applied to gene-expression data at different stages of cell cycle in yeast [3]. During recent years several algorithms, originating from various theoretical disciplines (e.g., physics, mathematics, statistics and computational neuroscience), were adopted and adjusted to gene expression analysis. They

[*] Corresponding author.

G. Chen et al. (Eds.): ISPA Workshops 2005, LNCS 3759, pp. 159 – 167, 2005.

are useful for diagnosis of different conditions for example differentiating between sick and healthy tissues, and classification to subtypes of a disease. An additional outcome of applying such algorithms to gene-expression data was the revealing of functional classes of genes among the thousands used in experimental settings [4]. Furthermore, it became possible, and useful, to isolate groups of relevant genes that mostly contribute to a particular condition, in the correlative or derivative perspective, a procedure called bi clustering [5].

By their nature, data points that are collected from large-scale experimental settings suffer from being represented in a high dimensional space. This fact presents a computational and an applicative challenge. Compression methods that maintain the fundamental properties of the data are called for.

As clustering algorithms are rooted in different scientific backgrounds and follow different basic principles, it is expected that different algorithms perform differently on varied inputs. Therefore, it is required to identify the algorithm that suits best a given problem. One of the targets of COMPACT is to address this requirement, and to supply an intuitive, user-friendly interface that compares clustering results of different algorithms.

In this paper we outline the key steps in using COMPACT and illustrate it on two well-known microarray examples of Leukemia [4], and yeast datasets [6]. For a comparative analysis we included routinely used clustering algorithms and commonly applied statistical tests, such as K-Means, Fuzzy C-Means and a competitive neural network. One novel method, Quantum Clustering (QC) [7], was added to evaluate its relative performance. The benefit of applying COMPACT to already processed data is demonstrated. All four algorithms that were applied in analyzing these datasets were compared with a biologically based validated classification. We conclude that the compression of data that comprises the first step in COMPACT, not only reduces computational complexity but also improves clustering quality. Interestingly, in the presented tested datasets the QC algorithm outperforms the others.

2 Implementation

After downloading and configuring COMPACT, four steps should be followed: defining input parameters, preprocessing, selecting the clustering method and presenting the results.

2.1 Input Parameters

COMPACT receives two input parameters that are Matlab variables: data (a two-dimensional matrix) – represents the elements to be clustered, and 'real classification' (an optional, one-dimensional vector) – representing the elements according to an expert view and is based on bulk biological and medical knowledge.

2.2 Preprocessing

a) Determining the matrix shape and which vectors are to be clustered (rows or columns).
b) Preprocessing Procedures: SVD, normalization and dimension selection.

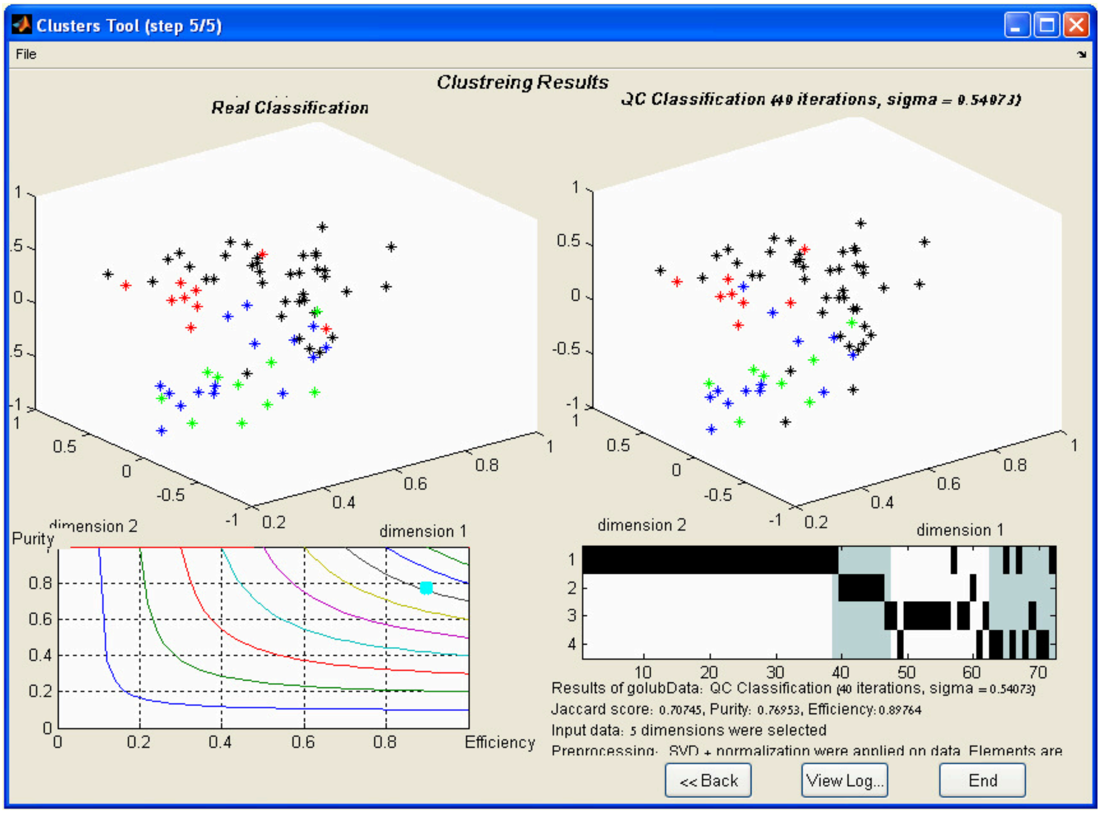
2.3 Selecting the Clustering Method

a) Points' distribution preview and clustering method selection: The elements of the data matrix are plotted. If a 'real classification' exists, each of its classes is displayed in a different color. One of the clustering methods, K-means, FCM (fuzzy C-means), Competitive NN (Neural Network) or QC (Quantum Clustering) is to be chosen from the menu.

b) Parameters for clustering algorithms: depending on the chosen method, a specific set of parameters should be defined (e.g., in the K-Means method – number of clusters).

2.4 COMPACT Results

Once COMPACT completes its run, the results are displayed in both graphical and textual formats (results can be displayed also in a log window). In the graphical display, points are tagged by the algorithm. The textual display represents Purity and Efficiency (also known as precision and recall or specificity and sensitivity, respectively) as well as the joint Jaccard Score[1]. These criteria for clustering assessment are defined as follow:



Results of golubData: QC Classification (40 iterations, sigma = 0.54073)
Jaccard score: 0.70745, Purity: 0.76953, Efficiency: 0.89764
Input data: 5 dimensions were selected
Preprocessing: SVD + normalization were applied on data. Elements are

Fig. 1. A screenshot of the graphical view on the results produced by COMPACT

[1] The Jaccard score reflects the ‘intersection over union' between the algorithm and ‘real’ clustering, and its values range from 0 (void match) to 1 (perfect match).

$$Purity = \frac{n_{11}}{n_{11} + n_{01}}, \quad Efficiency = \frac{n_{11}}{n_{11} + n_{10}}, \quad Jaccard = \frac{n_{11}}{n_{11} + n_{01} + n_{10}} \tag{1}$$

Where:

- n_{11} is the number of pairs that are classified together, both in the 'real' classification and in the classification obtained by the algorithm.
- n_{10} is the number of pairs that are classified together in the correct 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 correct classification.

Ending the application will add a new variable to the Matlab workspace: calcMapping - a one-dimensional vector that represents the calculated classification of the elements.

3 Results

We applied several of the most commonly used clustering algorithms for gene expression data. By analyzing the results of COMPACT we observe significant variations in performance. In the following we compare the performance on different datasets. We choose to use datasets that were heavily studied and for which an expert view is accepted.

3.1 COMPACT Tests of Leukemia Microarray Expression Data

We tested COMPACT on the dataset of Golub et al. [4] that has served already as a benchmark for several clustering tools (e.g. [2], [8], [9], [10], [11]). The experiment

Table 1. COMPACT based comparison for the Golub dataset [4]. For details see text.

Method	Jaccard	Purity	Efficiency
Raw data			
K Means	0.257	0.369	0.459
Fuzzy C Means (FCM)	0.272	0.502	0.372
Competitive Neural Network (NN)	0.297	0.395	0.547
Quantum Clustering (QC)	NA	NA	NA
Preprocessing (SVD)			
K Means	0.4	0.679	0.494
Fuzzy C Means (FCM)	0.316	0.584	0.408
Competitive Neural Network (NN)	0.442	0.688	0.553
Quantum Clustering (σ= 0.54)	0.707	0.77	0.898

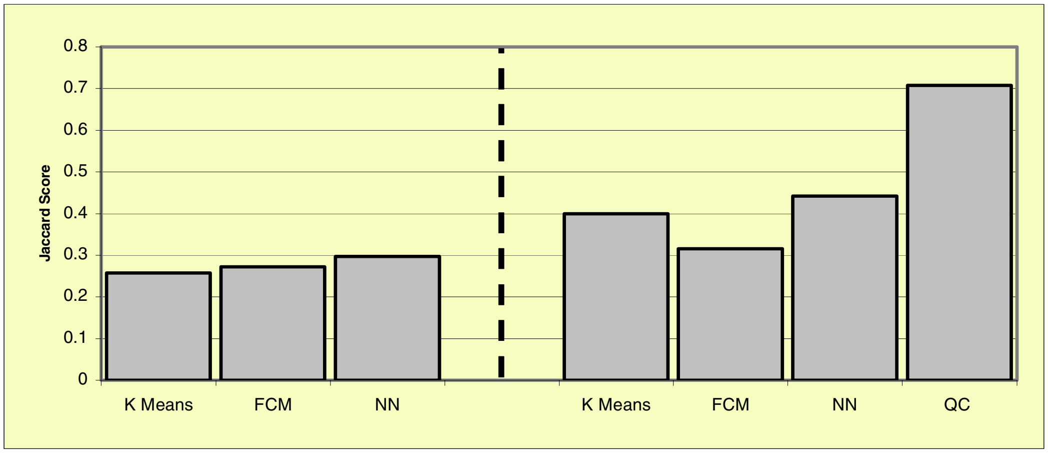


Fig. 2. Jaccard scores of the four algorithms tested by COMPACT on the Golub dataset. Left: before compression, Right: following application of the SVD compression step. Note that an improvement is detected for all methods by a preprocessing step.

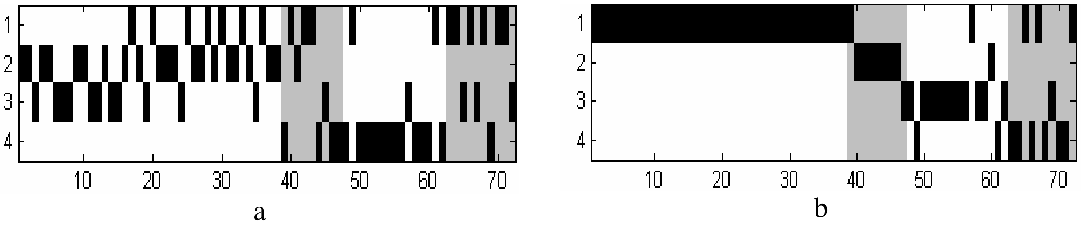


Fig. 3. A graphical comparison of COMPACT results on Leukemia dataset. The samples (patients) are ordered by their groups: samples 1-37: group #1, samples 38-47: group #2, samples 48-62: group #3 and samples 63-72: group #4. The four 'real' classes are distinguished by the background color (white, gray, white and gray), whereas black bars demonstrate the algorithm's classification. For example, in (a) the first sample belongs to the 'correct' first group (white background); while the algorithm placed it in the second group (the black bar is at group #2). Shown are the results of (a) K-means (K=4) and (b) QC (Quantum clustering, σ = 0.54) for clustering the AML/ALL cancer cells after SVD truncation to 5 dimensions.

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 chip measured the expression of 7129 genes.

The clustering results for four selected clustering algorithms are shown in Table 1. A comparison of the Jaccard scores for all algorithms is displayed in Figure 2 and two clustering assignments are compared in Figure 3. Applying the selected algorithms to the raw data (i.e., without an SVD preprocessing) yields poor outcomes.

Next we applied the SVD preprocessing step selecting and normalizing the 5 leading SVD components ('eigengenes' according to Alter, [12]) thus reducing the matrix from 7129X72 to 5X72. Clustering has improved after dimensional truncation, yet not all algorithms correctly cluster the samples. Note that only QC shows a substantial degree of consistency with the 'real' classification (Jaccard. = 0.707, Purity = 0.77 and Efficiency = 0.898, for discussion see Horn & Axel [13]).

3.2 COMPACT Tests of Yeast Cell Cycle Data

Next we test the performance of COMPACT for clustering of genes rather than samples. For this goal we explore the dataset of yeast cell cycle presented by Spellman et al. [6]. This dataset was used as a test-bed for various statistical and computational methods 14]. The expression levels of 798 genes were collected from 72 different

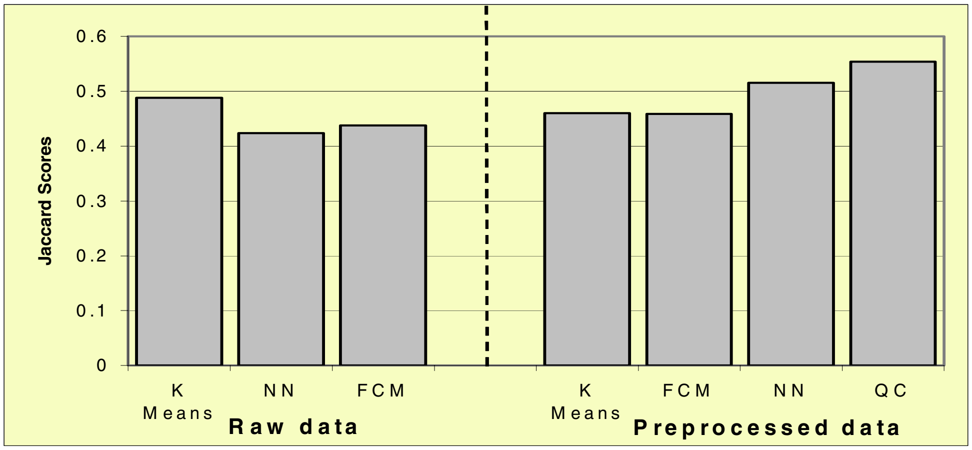


Fig. 4. Jaccard scores of the algorithms in the COMPACT based comparison for the Spellman dataset (shown are results for four clusters analysis)

Table 2. COMPACT based comparison to the Spellman dataset of Cell cycle in Yeast [6]

Method	Jaccard	Purity	Efficiency
Raw data			
K Means (5 clusters)	0.435	0.617	0.596
K Means (4 clusters)	0.488	0.64	0.673
Fuzzy C Means (5 clusters)	0.425	0.663	0.542
Fuzzy C Means (4 clusters)	0.438	0.458	0.912
Competitive Neural Network (4 clusters)	0.424	0.53	0.68
Quantum Clustering	NA	NA	NA
Preprocessing			
K means (5 clusters)	0.406	0.636	0.528
K means (4 clusters)	0.46	0.626	0.634
Fuzzy C means (5 clusters)	0.4	0.63	0.522
Fuzzy C means (4 clusters)	0.459	0.624	0.634
Competitive Neural Network (5 clusters)	0.33	0.55	0.458
Competitive Neural Network (4 clusters)	0.516	0.658	0.706
QC after SVD (σ =0.595)	0.554	0.664	0.77

conditions that reflect different time points in the yeast cell cycle. The task in this case is to cluster these 798 genes into five classes identified by Spellman et al. through functional annotations of individual genes.

We applied COMPACT both to ‘raw’ data and to SVD compressed data. In the latter case we selected two leading normalized SVD components ('eigensamples' according to Alter, [12]), thus reducing the matrix size from 798X72 to 798X2. All four clustering methods were tested as before. Once again the results obtained by the QC are moderately superior.

We have tested all methods for both 4 and 5 clusters (Table 2 and Figure 4). Interestingly enough, 4 clusters seem to be a better choice in all cases, although the 'real’ classification defines 5 classes.

4 Discussion

In this paper we demonstrate how different clustering algorithms may lead to different results. The advantage of COMPACT is in allowing many algorithms to be viewed and evaluated in parallel on a common test set. Through COMPACT one can evaluate the impact of changing the algorithm or its parameters (e.g., sigma value in QC, number of iterations for the Competitive Neural Network, starting points of K-Means, Fuzzy C-Means and more). Being able to run a number of clustering algorithms, observe their results (quantitatively and graphically) and compare between them is beneficial for researchers using gene expression, proteomics, and other technologies that produce large datasets. We find it advisable to start with a problem that has a known classification (referred to as ‘real classification’) and use the statistical criteria (i.e., efficiency, purity and Jaccard score) to decide on the favorable clustering algorithm. For general research problems, where no known classification exists, the same statistical tools may be used to compare results of different clustering methods with one another. We presented here a comparative analysis of some well-known clustering methods with one relatively new method, QC. For the two datasets that we have explored, QC outperformed the other methods.

We have shown that dimensionality reduction improves the clustering quality. This observation is highly relevant when handling genomic data. Recall that for Affymetrix microarrays the number of genes tested reaches all known transcripts from the selected organism, producing 20,000-30,000 data points for a mammalian genome. Similarly, the application of the new SNP discovery chip produces a huge number of noisy data points in a single experiment. Besides its computational complexity, one of the major challenges when using massive data is to identify features and to filter out noise. Often handling such high dimensional noisy inputs can be a barrier. Hence it is important to develop more efficient and accurate tools to tackle these problems (see examples in [3], [4], [15], [16]). Thus, constructing a method that can significantly reduce data volume, and at the same time keep the important properties of that data, is obviously required.

COMPACT offers easy-to-use graphical controls for users to select and determine their own preferences, and graphical displays where the results can be presented or saved for later usage. It offers several clustering algorithms and allows the user to compare them to one another.

Although similar tools have already been proposed (e.g., [17], or [18]), the novelties of COMPACT are: (i) presenting an integrative, light package for clustering and visualization, (ii) integrating an efficient compression method and (iii) introducing the QC algorithm as part of the available clustering options.

The beginners will find this user-friendly tool with its graphical and textual displays useful in their data analysis. The experts will benefit from its flexibility and customizability that enables expanding the tool and modifying it for advanced, specialized applications.

Acknowledgment

We thank Inon Axel for helping in the initial stages of the development. We thank Noam Kaplan for critical reading the manuscript. We thank the ProtoNet team in the Hebrew University of Jerusalem for suggestions and helpful discussions. This work is partially supported by the BioSapiens NoE of the EU framework VI. R.V. awarded a fellowship by the SCCB, The Sudarsky Center for Computational Biology. His research is partially supported by the Israel Science Foundation.

Availability: COMPACT is available at http://www.protonet.cs.huji.ac.il/compact and at http://adios.tau.ac.il/compact . A detailed description of the application can be found on these websites.

References

1. Jain, A. K., Dubes R. C.: Algorithms for Clustering Data. Englewood Cliffs, NJ, Prentice Hall, Englewood Cliffs, NJ; 1988
2. Sharan, R., Maron-Katz A., Shamir, R.: CLICK and EXPANDER: a system for clustering and visualizing gene expression data. Bioinformatics. 2003, 19(14): 1787-99.
3. Eisen, M. B., Spellman, P. T., Brown P. O., Botstein D.: Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A. 1998, 95(25): 14863-14868.
4. 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., Lander E.S.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science. 1999, 286: 531-537.
5. Cheng Y., Church G. M.: Biclustering of Expression Data. Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology; AAAI; 2000:93-103.
6. Spellman P.T., Sherlock G., Zhang M.Q., Iyer V.R., Anders K., Eisen M.B., Brown P.O., Botstein D., Futcher B. P. T.: Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell. 1998, 9(12): 3273-97.
7. Horn, D., Gottlieb A.: Algorithm for data clustering in pattern recognition problems based on quantum mechanics. Phys Rev Lett. 2002, 88(1): 018702.
8. Yeang C.H., Ramaswamy S., Tamayo P., Mukherjee S., Rifkin R.M., Angelo M., Reich M., Lander E., Mesirov J., Golub T. C. H., Ramaswamy S.: Molecular classification of multiple tumor types. Bioinformatics. 2001, 17 Suppl 1: S316-22.

9. Pan, W.: A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. Bioinformatics. 2002, 18(4): 546-54.
10. Mukherjee S., Tamayo P., Rogers S., Rifkin R., Engle A., Campbell C., Golub T.R., Mesirov J.P. S.: Estimating dataset size requirements for classifying DNA microarray data. J Comput Biol. 2003, 10(2): 119-42.
11. Pan, W.: A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. Bioinformatics. 2002, 18(4): 546-54.
12. Alter, O., Brown P. O, Botstein D.: Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci U S A. 2000, 97: 10101-10106.
13. Horn, D., Axel I.: Novel clustering algorithm for microarray expression data in a truncated SVD space. Bioinformatics. 2003, 19(9): 1110-5.
14. Friedman, N., Linial M., Nachman I., Pe'er D.: Using Bayesian networks to analyze expression data. J Comput Biol. 2000, 7: 601-20.
15. Sasson, O., Linial N., Linial M.: The metric space of proteins-comparative study of clustering algorithms. Bioinformatics. 2002, 18 Suppl 1: S14-21.
16. Sasson O., Vaaknin A., Fleischer H., Portugaly E., Bilu Y., Linial N., Linial M.: ProtoNet: hierarchical classification of the protein space. Nucleic Acids Res. 2003, 31(1): 348-52.
17. The Eisen Lab software page [http://rana.lbl.gov/EisenSoftware.htm]
18. The R project for statistical computing [http://www.r-project.org/]

Section 4.3

ClusTree: A Simple Graphical Tool for Analysis of Hierarchical Clustering

Abstract

Summary: ClusTree is a graphical tool that enables an easy and intuitive way to apply, analyze, visualize and compare various hierarchical clustering methods. This expandable, Matlab package can either apply hierarchical clustering to experimental datasets (e.g., gene-expression), or visually and statistically evaluate trees which resulted from any hierarchical algorithms.

An obvious strength of the ClusTree tool is its capability to easily apply numerous algorithms to different inputs, and thus to be utilized for a wide range of data, ranging from gene-expression to proteins or nucleic acids sequences.

1. Introduction

Clustering is a common procedure in genomic and proteomic studies. Clustering algorithms are classified as either nonhierarchical (flat, partitioning) or hierarchical. While the former define a single partition of the data (e.g., K-Means), hierarchy, by its nature, suggests multiple levels of organization (for comprehensive reviews see Jain and Dubes, 1988; Duda, et al., 2000; D'Haeseleer, 2005).

The results of hierarchical clustering can be represented as a tree, where each grouping of nodes may define a cluster. A collection of nodes may be viewed as natural cuts in the tree. Cutting the tree can be done at different heights, which are in essence, provide multiple clustering solutions. For this reason, hierarchical clustering is usually considered as a richer organization method than nonhierarchical clustering. Some of the clustering possibilities may match an expert's view, while others may identify clusters that were not previously recognized as such.

Hierarchical clustering has been successfully applied to protein sequences (Sasson, et al., 2003), chemical entities, ontologies, 3D structural information, protein catalytic activities (Handl, et al., 2005), and in many large-scale gene expression experiments (Spellman, et al., 1998; D'Haeseleer, 2005). They have been implemented in applications such as (Eisen, et al., 1998; Saldanha, 2004; MathWorld, 2007), ClusTr (Apweiler, et al., 2001) and ProtoNet (Sasson, et al., 2003).

We present a new, intuitive graphical tool, named ClusTree, with novel improved capabilities. ClusTree is a very simple user-friendly collection of several algorithms. In addition to the standard agglomerative procedures (MathWorld, 2007), it provides an access to advanced top-down algorithms. In addition, introducing an array of statistical routines and visualization options it may assist in evaluating and comparing clustering results. Furthermore, being a self-explanatory graphical application, ClusTree can be easily comprehended by non-computational users; yet by suggesting an advance mode, it can be expanded by more sophisticated analysts. Finally, being a generic toolbox it is capable of handling datasets from various domains (e.g., gene-expression, sequence analysis).

2. Functionalities Provided

2.1. ClusTree Workflow

Figure 1 displays the workflow of the ClusTree tool. ClusTree can either cluster a given experimental dataset (input A, see 2.2) or visualize and analyze a dataset that has already been clustered (input B).

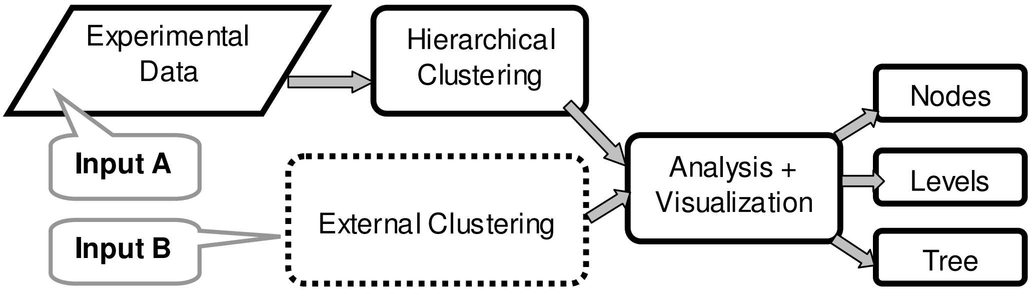


Fig. 1. ClusTree workflow: input (section 2.2), clustering (2.3), analysis and visualization 2.4.-2.5 and 2.6, respectively).

2.2. Input types

ClusTree is a generic tool that accepts two types of input: experimental data in feature-space (i.e., raw data as in most gene-expression experiments) or distance matrices, in which each element is the distance (or similarity) between two instances (e.g., pairwise distances from BLAST, marked by their statistical significance, e-values (Altschul, et al., 1997)).

2.3. Clustering options

The tool includes an interface to the standard hierarchical procedures provided in the statistical toolbox of Matlab. There are 10 common distance measures (euclidean, seuclidean, cityblock, mahalanobis, minkowski, cosine, correlation, hamming, jaccard and chebychev), and 7 linkage options (single, complete, average, weighted, centroid, median and ward). Altogether there are 70 combinations that are applicable. In addition, we provide two top-down hierarchical algorithms: PDDP (Boley, 1998) and TDQC, that were shown to be very effective in comparison to bottom-up ones (see Chapter 3).

2.4. Clustering evaluation

After the dataset is clustered, it can be visualized and analyzed. When expert-classification is provided (the term 'expert' refers to external data labelling, e.g., GO annotation (Camon, et al., 2004), we apply three combined assessment methods to describe different qualities of the clustering tree.

(1) Node Score: Each node specifies a cluster (of all its descendants). An enrichment *p*-value is calculated to assign any node with one of the classes in the data. This is done by using the hypergeometric probability function (Rivals, et al., 2007). The *p*-values for all nodes may be viewed as dependent set estimations, hence we apply the False Discovery Rate (FDR) criterion to them (Benjamini and Hochberg, 1995). *p*-values which fail to pass this criterion are considered not significant. An additional criterion is also provided: a node is considered significant only if a certain fraction (default is 50%) of its elements belongs to the enriched category.

(2) Level Score: Level l of the tree includes all clusters that are l edges away from the root. Choosing for each node the class for which it turned out to have a significant node score, we evaluated its Jaccard-score *(J=tp/(tp+fn+fp),* where *tp* is the number of true positive cases, *fn* the number of false negative cases and *fp* the number of false positive cases) (Sharan, et al., 2002; Varshavsky, et al., 2005). If the node in question is not significant by the enrichment criterion, its J-score is set to null. The level score is defined as the average of all J-scores at the given level.

(3) Tree Score is the weighted best-J-Score, $J^* = \frac{1}{N}\sum_i^c n_i J_i^*$, where J^*_i is the best J-Score for class i in the tree, n_i is the number of instances in class i, c is the number of classes and N is the number of instances in the dataset. This score or its close variation has been applied to measure the quality of protein families (Kaplan, et al., 2005) and document classification (Steinbach, et al., 2000; Zhao and Karypis, 2002).

2.5. Additional scoring options

Alternative tree scores calculated by the tool are the C and F scores:

C Score is the relative number of significant nodes (# significant nodes/ # nodes in the tree).

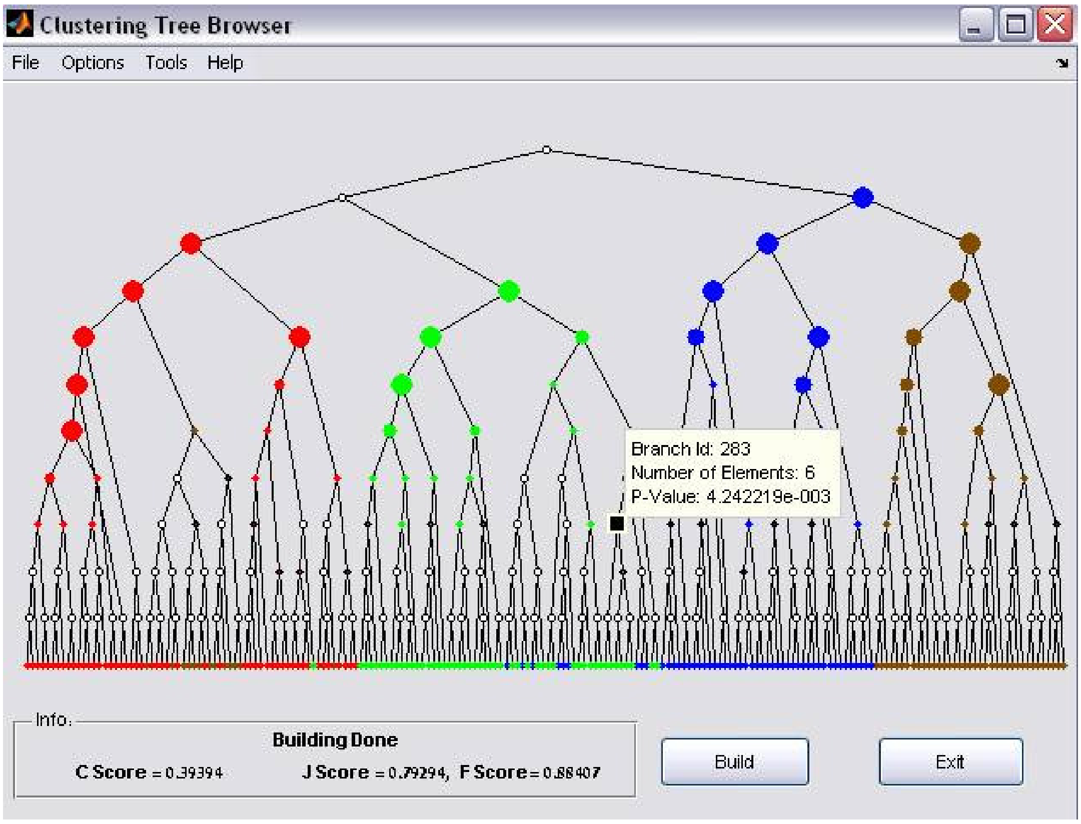
F Score is analogous to the J^* tree score, defined as the weighted best-F-Score: $F^* = \frac{1}{N}\sum_i^c n_i F_i^*$, where F^*_i is the best F-Score, $F - Score = \frac{2 * recall * precision}{recall + precision}$,

where $recall = \frac{tp}{tp + fn}$, $precision = \frac{tp}{tp + fp}$.

For class i in the tree, n_i is the number of data-points in class i, c is the number of classes and N is the number of data-points in the dataset.

2.6. Clustering visualization

Figures 2 and 3 display examples of ClusTree graphical outputs. Displayed in Figure 2 is a hierarchy tree example. The tree is colored according to the node scores. A node size is proportional to its statistical-enrichment level. In addition, selecting a node (by clicking it, Figure 2) further analysis can be performed (e.g., querying included instances).



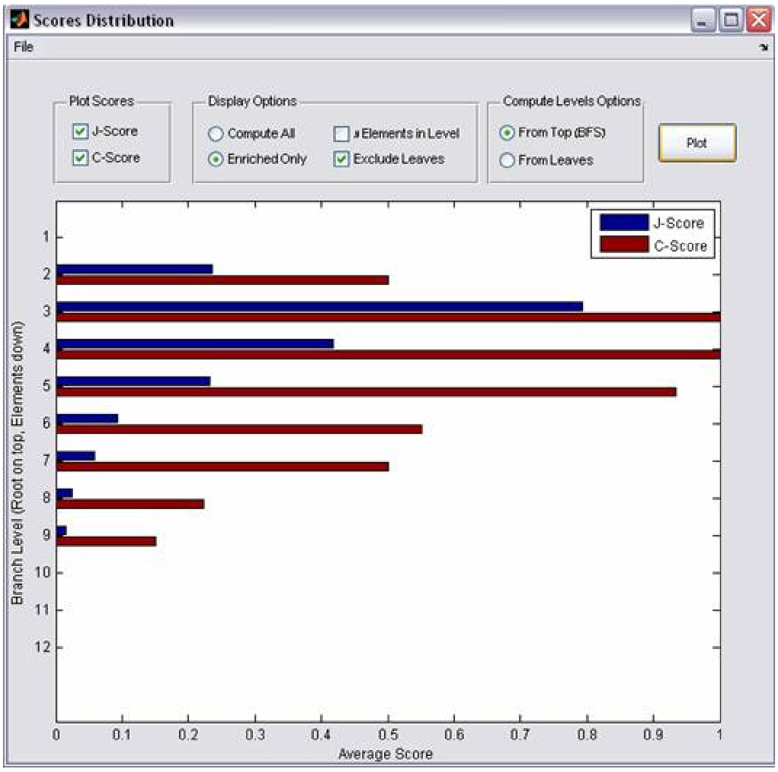
Branch Id: 283
Number of Elements: 6
P-Value: 4.242219e-003

Building Done

C Score = 0.39394 J Score = 0.79294, F Score = 0.88407

Fig. 2. A screenshot of the ClusTree results. Dot sizes indicate statistical enrichment levels where larger sizes correspond to smaller, more significant *p*-values. Empty nodes represent no enrichment. The black square is a clicked node whose properties are quoted in the tool-tip.

A different perspective is provided by the level scores display (Figure 3), allowing a condensed view of the distribution of the significant nodes along the tree.



Plot Scores

☑ J-Score

☑ C-Score

Display Options

○ Compute All

● Enriched Only

□ # Elements in Level

☑ Exclude Leaves

Compute Levels Options

● From Top (BFS)

○ From Leaves

Fig. 3. A screenshot of the level scores window. Presented are J and C scores for each tree level. As shown, the best clustering quality is observed at level 3.

2.7. Additional functionalities

In addition to the standard routines, some advanced functionalities are available (detailed descriptions provided in the manual).

Ultrametric view: a tree, whose edges are viewed as discrete integer distances, dictates an ultrametric space hosting the data points. This representation can be observed, exported and studied.

Export/save options: a tree and its graphical representation can be saved either as a Matlab variable, text file or as a figure.

Expand the toolbox: the software was designed so that adding new algorithms can be easily done.

3. Conclusions

Hierarchical clustering is a routinely used strategy. We provide simple, intuitive software for applying various hierarchical clustering algorithms, and analyzing their results. Analysis can be performed both in a quantitative way, by scoring different resolutions within the tree, and in a qualitative way, by visualizing the resulting trees. These methods allow for straightforward and comprehensive comparisons between competing and complementing algorithms.

Acknowledgements

We thank Alon Kaufman, Nati Linial, Assaf Gotlieb, Yaniv Loewenstein and the ProtoNet team for stimulating discussions and theoretical assistance. We thank Liat Segal for graphical assistance. R.V. is supported by SCCB, the Sudarsky Center for Computational Biology at the Hebrew University of Jerusalem. This study was supported by the EU Framework VI NoE DIAMONDS consortium.

References

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res*, **25**, 3389-3402.

Apweiler, R., Biswas, M., Fleischmann, W., Kanapin, A., Karavidopoulou, Y., Kersey, P., Kriventseva, E.V., Mittard, V., Mulder, N., Phan, I. and Zdobnov, E. (2001) Proteome Analysis Database: online application of InterPro and CluSTr for the functional classification of proteins in whole genomes, *Nucleic Acids Res*, **29**, 44-48.

Benjamini, Y. and Hochberg, Y. (1995) Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, *Journal of the Royal Statistical Society. Series B (Methodological)*, **57**, 289-300.

Boley, D. (1998) Principal Direction Divisive Partitioning. *Data Mining and Knowledge Discovery*. 325 - 344.

Camon, E., Barrell, D., Lee, V., Dimmer, E. and Apweiler, R. (2004) The Gene Ontology Annotation (GOA) Database--an integrated resource of GO annotations to the UniProt Knowledgebase, *In Silico Biol*, **4**, 5-6.

D'Haeseleer, P. (2005) How does gene expression clustering work?, *Nat Biotechnol*, **23**, 1499-1501.

Duda, R.O., Hart, P.E. and Stork, D.G. (2000) *Pattern Classification*. Wiley-Interscience.

Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns, *PNAS*, **95**, 14863-14868.

Handl, J., Knowles, J. and Kell, D.B. (2005) Computational cluster validation in post-genomic data analysis, *Bioinformatics*, **21**, 3201-3212.

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

Kaplan, N., Sasson, O., Inbar, U., Friedlich, M., Fromer, M., Fleischer, H., Portugaly, E., Linial, N. and Linial, M. (2005) ProtoNet 4.0: a hierarchical classification of one million protein sequences, *Nucleic Acids Res*, **33**, D216-218.

MathWorld (2007) Matlab Statistics Toolbox. MathWorld.

Rivals, I., Personnaz, L., Taing, L. and Potier, M.-C. (2007) Enrichment or depletion of a GO category within a class of genes: which test?, *Bioinformatics*, **23**, 401-407.

Saldanha, A.J. (2004) Java Treeview--extensible visualization of microarray data, *Bioinformatics*, **20**, 3246-3248.

Sasson, O., Vaaknin, A., Fleischer, H., Portugaly, E., Bilu, Y., Linial, N. and Linial, M. (2003) ProtoNet: hierarchical classification of the protein space, *Nucleic Acids Res*, **31**, 348-352.

Sharan, R., Elkon, R. and Shamir, R. (2002) Cluster analysis and its applications to gene expression data, *Ernst Schering Res Found Workshop*, 83-108.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. (1998) Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization

Mol. Biol. Cell, **9**, 3273-3297.

Steinbach, M., Karypis, G. and Kumar, V. (2000) A comparison of document clustering techniques. *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Boston.

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

Zhao, Y. and Karypis, G. (2002) Evaluation of hierarchical clustering algorithms for document datasets. In, *Proceedings of the eleventh international conference on Information and knowledge management* ACM Press, McLean, Virginia, USA 515-524

Chapter 5

When Less is More: Improving Classification of Protein Families with a Minimal Set of Global Features

This chapter contains the following research paper:

[5A] **Roy Varshavsky**, Menachem Fromer, Amit Man and Michal Linial. "*When Less is More: Improving Classification of Protein Families with a Minimal Set of Global Features*" (2007, WABI, Lecture Notes in Computer Science (4645), 12-24).

When Less Is More: Improving Classification of Protein Families with a Minimal Set of Global Features

Roy Varshavsky[1,**], Menachem Fromer[1], Amit Man[1], and Michal Linial[2]

[1] School of Computer Science and Engineering, The Hebrew University of Jerusalem
[2] Department of Biological Chemistry, The Hebrew University of Jerusalem
royke@cs.huji.ac.il

Abstract. Sequence-derived structural and physicochemical features have been used to develop models for predicting protein families. Here, we test the hypothesis that high-level functional groups of proteins may be classified by a very small set of global features directly extracted from sequence alone. To test this, we represent each protein using a small number of normalized global sequence features and classify them into functional groups, using support vector machines (SVM). Furthermore, the contribution of specific subsets of features to the classification quality is thoroughly investigated. The representation of proteins using global features provides effective information for protein family classification, with comparable results to those obtained by representation using local sequence alignment scores. Furthermore, a combination of global and local sequence features significantly improves classification performance.

Keywords and Abbreviations: Support Vector Machines (SVM), Feature Selection, Olfactory Receptor, Porins protein family.

1 Introduction

Protein classification is a central task in computational biology. A routinely-used principle in classification relies on a distance measure between protein sequences, as obtained by the Smith-Waterman local alignment algorithm or by one of a large number of heuristic search methods such as BLAST, PSI-BLAST [1], search by HMM [2, 3] models and by profile-profile search [4, 5]. These methods are typically based on matching subsequences, i.e. local sequence features.

Despite the observed strength of these methods, many functional assignments for proteins fail to be detected by such local sequence-based methods [6], thus

[**] Corresponding author.

We thank David Horn for advising and guiding R.V, Nati Linial, Elon Portugaly and Yaniv Loewenstein for fruitful discussion. R.V and M.F are supported by the Sudarsky Center for Computational Biology of the Hebrew University of Jerusalem. This research was partially supported by a grant from the Israel Ministry of Defense.

Supplementary Data and Code: www.protonet.cs.huji.ac.il/sequence_features.

R. Giancarlo and S. Hannenhalli (Eds.): WABI 2007, LNBI 4645, pp. 12–24, 2007.

yielding a larger than desired fraction of false negatives, especially at more coarse-grained (higher) levels of protein classification hierarchies. The shortcomings of the methods outlined above are partly derived from the fact that there exist many proteins that share very low sequence similarity and are thus considered to be in the ”twilight zone”, but nonetheless share strong structural similarity that reflects their homology [7]. Short proteins represent another set of proteins that often fail to be classified by their sequence similarity due to their low statistical significance scores [8]. Finally, for many proteins the sequence similarity methods fail in detecting related sequences and as a result, a large fraction of singletons are reported within the protein space [9].

An additional confounding factor is that, in practice, the large number of protein sequences currently available imposes a computational challenge for the protein family classification problem. Currently, $>$ 4.5 million sequences are stored in the UniProt database, and this collection is expected to grow [10]. A reduction to 3 and to 1.5 million sequences is achieved by UniRef90 and UniRef50, respectively (i.e., no two sequences are permitted to share more than 90% or 50% identity, respectively). Since even such vast reductions in redundancy yield very large quantities of sequences, the power of the ubiquitously used local sequence similarity methods are severely strained. Similarly, each new multi-cellular eukaryotic genome sequenced introduces thousands of new sequences that wait for functional assignments, again burdening the local sequence similarity algorithms.

To address the challenges in large-scale functional assignment, a complementary line of research has used a spectrum of sequence features ranging from amino acid (aa) composition to the appearance of short sequence motifs [11]. Besides perhaps improving upon the results of local-based methods, this research is expected to provide information for classification of more distantly related protein families, where local-based methods may often fail. One such attempt was presented by SVM-Prot [12]. The classification system was trained from representative proteins for $\sim$50 functional families extracted from Pfam [13]. Using a large number of features and an SVM classifier, high success in separating these protein families was reported. A different approach was carried out in [14], where a mixture of probabilistic decision trees for direct prediction of protein functions was applied. In [14], the proteins are represented by hundreds of features, including secondary structure assignment and structural-based information.

Despite their success, these approaches do not always allow for interpretations and inferences based on the full interplay among features. In addition, the large set of features used could inadvertently conceal the fact that the prediction task is easier than it seems: it may be sufficient to consider only a small set of global features. While it may seem overly ambitious to expect the task of protein family classification to succeed based only on a small set of sequence features, similar features were successfully applied for restricted, but related, tasks. Successful examples include distinguishing membranous and globular proteins, separating sub-cellular localization, [15], determination of topology for multi-pass proteins [16], and even prediction of protein quaternary structure [17].

Herein we assume a minimalist feature-based approach, which for reductionism-based motivations does not take into account secondary or tertiary structure information, even when reliable predictions are available. Moreover, we ignore features derived from short motifs that are currently known to be associated with specific protein families, functions, or subcellular localizations. We thus address the following questions regarding a small set of easily extracted global sequence features: *(i)* Does there exist a small (minimal) set of features that provides high-quality protein family characterization? *(ii)* Is the information conveyed by global features redundant or, rather, complementary to that provided by the local features? *(iii)* And, more generally, are there some biological insights that predict the prototypical successes and failures of feature-based classifications?

To define the minimal set of features sufficient for functional classification, we: *(i)* test the capacity of predetermined, small subsets of features, and *(ii)* incorporate machine learning tools (specifically, feature selection) to automatically determine those features. Feature selection is a fundamental component in large-scale data analysis as a preprocessing step. In general, preprocessing involves some operation on the feature-space intended to reduce the dimensionality. In feature selection, only a particular subset of features is chosen and used in subsequent computational tasks. There are two major classes of feature selection strategies: filters and wrappers. Filter methods rank and choose the features according to some criterion (e.g., data separation). Wrapper methods optimize an objective function, through the selection of features. For a comprehensive survey, see [18]. Herein, we apply one filter and two wrappers to the data.

2 Data and Methods

2.1 Data

As a test case, we consider 10 large protein groups that represent the known diversity of cellular processes and functions. Protein sequences and annotationss were retrieved from the UniProt 8.1 database [10]. In order to avoid redundancy, we used the UniRef50 database [10]. Groups were selected based on Gene Ontology (GO) assignments [19], such that their sizes would range from 300–1000 proteins each. 5,471 proteins in total are included in the analysis (Table 1).

2.2 Preprocessing

We compare two alternative representations of these ∼5,500 proteins: either according to local sequence similarities, or according to global sequence features:

1. Local Sequence similarities
   All pairs of proteins were aligned using the Smith-Waterman (SW) local alignment algorithm [20]. Since the SW score is strongly dependent on protein length, the raw scores matrix was transformed to a matrix of normalized scaled scores, based on the percentile binning of scores in each column. As a result, the range of values in the scaled matrix is $[0, 1]$. Note that the column-by-column transformation yields an asymmetric matrix.

Table 1. Representative set of 10 groups derived from the GO systems: cellular component (CC), molecular function (MF) and biological process (BP)

Group	**Type**	**CO Term name**	**GO ID**	**Group Size (UniRef50)**
1	CC	Nucleosome	786	319
2	MF	Olfactory receptor activity	4984	478
3	CC	Vacuole	5773	533
4	CC	Microtubule	5874	913
5	CC	Plasma membrane	5886	781
6	BP	Tricarboxylic acid cycle	6099	476
7	BP	DNA unwinding duringreplication	6268	520
8	CC	Thylakoid	9579	448
9	MF	Porin activity	15288	644
10	CC	Myosin complex	16459	359
Total				**5471**

2. Global Sequence Features
   Extracting the features: Only features that are "global" and can be applied to proteins with minimal biological pre-knowledge are included (e.g., the calculated isoelectric point of a protein). Biologically known signatures such as localization signals were not included. In summary, for each protein, 5 major attribute types (for a total of 70 features) are analyzed:

   Amino acid composition [AAC] (20 features).
   Amino acid grouped compositions [AAG] (11 features, see Table 3, Supplementary Data).
   Post-translational modifications [PTM] (14 features, see Table 4, Supplementary Data). The PTM signatures are treated as regular expressions. Such patterns have been extracted from the Prosite database [21]. Only PTMs that are highly abundant in the database are included.
   Biophysical properties of the full sequence [PHYS] (5 features):

   (a) Length - The number of amino acids in the sequence
   (b) Molecular weight [22]
   (c) predicted pI [22]
   (d) Instability factor: based on the observation that the frequency of occurrence of certain dipeptides is significantly different in unstable proteins as compared to stable ones [23].
   (e) 'Gravy' hydrophobicity index [24]

   Amino acid enrichment [RICH] (20 features). We sampled an overlapping window of 20 aa in size, from the beginning of the sequence to the end. For each such window, the frequency of a certain aa was counted if it occurs at least 5 times its frequency in the UniProtKB database.

Scaling the features: Since the selected features represent properties that appear in vastly different representations (e.g., logarithmic scale for pI, percentage for AAC, frequency for RICH), we applied a scaling protocol by referring to a background level of a randomly selected set of approximately 40K proteins from the UniProtKB database. For each of the 70 features the percentile bins of the background were computed. Each feature was transformed according to its percentile, yielding values in the range $[0, 1]$. We also applied the scaling using a background set of the 5,500 proteins in our set (Table 1) and the results were practically identical to that of the randomly selected background set.

2.3 Classification

Firstly, the 10 groups were randomly partitioned into 3 subsets (groups 1-4, 5-7, and 8-10), where it was attempted to separate each group of proteins from the other groups in its subset. The classification algorithm chosen for the task was SVM (linear kernel, one-against-all classification), which has been proven to be very efficient for this type of task (e.g. [12, 11]). For each dataset in every representation used, the following procedures were applied:

1. Random selection of the train (80%) and test (20%) sets.
2. Use the train set: train and validate SVM (5-fold Cross validation).
3. Apply the resulting classifier to the test set, for prediction and assessment.

In order to reduce bias toward extreme train-test partitions, procedures 1-3 (which we refer to as the *classification block*) were repeated 5 times (which we refer to as the *classification compound*).

2.4 Feature Selection

We consider two strategies for selection of the global sequence features, applying the *classification compound* for each. Note that the selections and wrappings are applied only to the train set.

- Selection based on a-priori knowledge. The original (scaled) dataset is partitioned according to the 5 different feature categories: AAC (20), AAG (11), PTM (14), PHYS (5) and RICH (20).
- Supervised feature selection methods. Here, various approaches are applied:
  1. Single-wise selection (GREEDY) – a filter method: the 70 features in the train set are ranked according to their t-test separability criterion – the first 10 features are selected.
  2. Forward Filtering (FF) – a wrapper method, which starts out with 0 features and adds the most contributing feature to the predictive score (Jaccard, see below) of the train set. Feature addition is continued until no improvement in the score is achieved.
  3. Backward Elimination (BE) – a wrapper method, which starts out with all features and removes the least contributing feature to the predictive score (Jaccard, see below) of the train set. Feature removal is continued until no improvement in the score is achieved.

2.5 Evaluation

For each classification block, TP, TN, FP, and FN counts are recorded (where TP, TN, FP, and FN denote the number of true positive, true negative, false positive, and false negative outcomes, respectively (detailed tables of all values appear in the Supplementary Data). We have applied the strict Jaccard score (J-score) that combines precision (specificity) and recall (sensitivity), but does not take into account the TN. The J-score is defined as: J = TP/(TP+FP+FN).

3 Results

In order to demonstrate both the strengths and limitations of the framework, we describe the results for two example groups. Detailing both computational and biological aspects, we demonstrate different scenarios that directly derive from the groups' characterization (for the remaining 8 groups, see Supplementary Data); we then discuss the overall patterns, suggest a unique feature combination platform and draw some conclusions. We analyzed large sets of proteins based on their GO annotations. For representative sets, we ensured that their sizes (at a level of lower than 50% identity for any pair in the set) ranged from 300-1000 and that, overall, they represent a broad range of functionality of enzymes, membranous components (olfactory and transporters), cytoskeletal elements (myosin) and compartment-based annotations (i.e. vacuole).

3.1 Olfactory Receptor Activity Proteins

The first group we consider is the olfactory receptor activity proteins, consisting of ∼500 proteins (3,900 proteins in UniProtKB), which are cell surface receptors that recognize chemical compounds (odorants). Odorant binding to its cognate receptor leads to membrane depolarization, activating a signaling cascade.

Could we gain any insight into the group, by revisiting the features selected to separate it from the other groups tested? Here, the FF approach performs almost as well as using all features (0.89 and 0.91, respectively, Fig. 1). Only 8 features are chosen by FF: AAG (hydrophilic), AAC (G), RICH (Y), PHYS (instability), AAC (T), AAG (sulfur-containing), AAC (V), and AAG (helix-redundant aa).

The most powerful feature selected under the FF protocol marks the hydrophilic nature of this protein group. Even though the olfactory receptors are characterized by their seven membrane-transversing helices, the hydrophobic nature of these helices was not among the separating features. On the other hand, the leading feature chosen was the hydrophilic signal of the molecule, derived from the region of the protein facing the aqueous environment on either side of the membrane (protein loops and tails). In an effort to characterize motifs that specify the olfactory receptors, 10 short motifs were determined, and they were all found to reside in the loops and tails of the proteins [25]. Similarly, 5 short PSSM motifs were used to characterize this family by BLOCKS [26]. Again, four of them are indeed in the hydrophilic segments of the proteins.

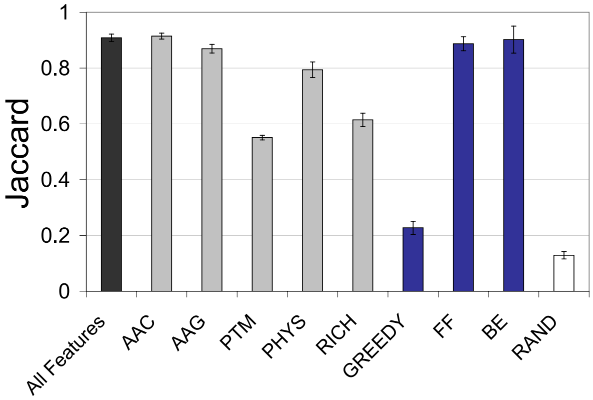


Fig. 1. J-score results of SVM classification, for various protein representations, of the olfactory receptor activity group. Bars are of all 70 global features (All: black), the 5 different feature types (AAC, AAG, PTM, PHYS and RICH: gray), and the 3 automated feature selection schemes (GREEDY, FF and BE: blue). As a reference, a random classification of the dataset is shown (100 iterations, RAND: white).

Other features yielded by FF include the frequency of glycine (G) and threonine (T). Also, among the features that contributed to separation is the richness of tyrosine (Y). It has been noted that tyrosine is quite abundant, and specifically a short sequence of 'MAYDRY' (tyrosine at positions 3 and 6) is conserved among most of the olfactory receptors in the group [27]. This short sequence has led to significant enrichment over the entire tested set. The rest of the selected features are cysteine (C) and methionine (M) (grouped as sulfur-containing aa), valine (V), and, the helix redundant amino acids group. The fact that this group of transmembrane proteins was distinguished from the other groups through the use of helix redundant amino acids is not completely surprising, since the proteins' membrane-spanning segments are composed of alpha helices. This detailed example illustrates that the selection of the most informative features (8 features in this case) covers diverse but complementary properties of the proteins.

3.2 Porin Proteins

The other group we discuss is that of bacterial porin, consisting of about 650 proteins (3,500 proteins in UniProtKB) that are localized to the outer membrane of Gram-negative bacteria, but also found in plastidae and mitochondria [28]. As one of the major outer membrane proteins in bacteria, they form large channels that allow the diffusion of small hydrophilic molecules ($<$ 1000 daltons). Classification results for the porin proteins group are displayed in Fig. 2.

Classification quality reaches a J-score of ~0.75. The global feature methods outperform the local feature method (J-score ~0.66). Interestingly, FF requires only three features for successful classification (J-score 0.68): AAC (G), AAC (I), and AAG (aromatic). To evaluate the relative contribution of each of these features, we have applied the *classification compound* using either the first 1, 2 or 3 features. The results (Fig. 3) show that the first feature by itself has a strong

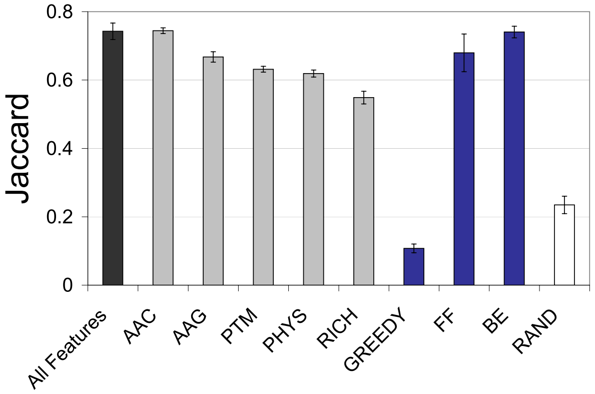


Fig. 2. Results of SVM classification, for various protein representations, of the porin activity proteins (notations, axes and colors are as in Fig. 1)

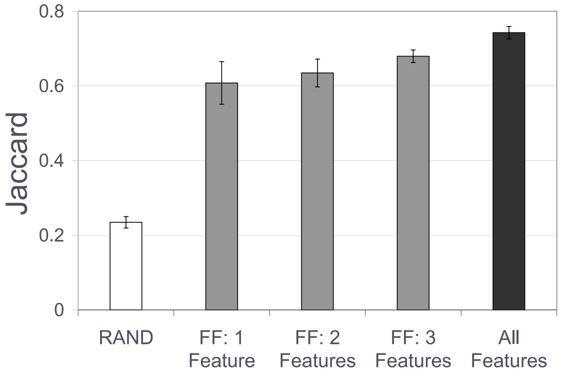


Fig. 3. The contribution of the first 3 features, selected by the FF method, to the classification quality of the porins group. The results are of random classification (white), classification using the single most, two most, and three most contributing features (AAC (G), AAC (I), and AAG (aromatic), gray), and all 70 features (black).

classification capability, with marginal contributions by the following two. The remaining 67 features have only a negligible contribution.

3.3 Group Size, Selection Method and Success

In order to estimate which protein families are best characterized by global features and which methods are preferred, we have applied several analyses. We computed the number of selected features in BE and FF. For the 10 groups of proteins presented here, the average number of features *eliminated* in the BE protocol is 5.4, and for FF an average of 5 features were selected. The extreme cases for the FF are the 3 features of the Porin group and 8 features for the olfactory protein group. These numbers and the average success in classification show no correlation with the number of proteins in the group (not shown).

Next, we compare the various selection methods. The scores for the selection methods are displayed in Table 2. As shown, the selection method that yields

Table 2. Average and standard deviation of the classification scores, according to the various selection methods

Selection Method	Number of Features	Average J-score	J-score StDev
All	70	0.67	0.126
AAC	20	0.63	0.149
AAG	11	0.57	0.188
PTM	14	0.45	0.171
PHYS	5	0.52	0.185
RICH	20	0.45	0.148
GREEDY	10	0.26	0.150
FF	5	0.56	0.163
BE	64.6	0.65	0.126

the highest scores is BE, followed by AAC (average J-scores 0.65 and 0.63, respectively). Not surprisingly, however, these are also the ones that retain high numbers of features (64.6 and 20, respectively). Nevertheless, it is noteworthy that the FF method yields a relatively high average score (J-score 0.56), although it uses as few as 5 features, on average. Another observation is that the more features selected, the lower the standard deviation of the J-score; this suggests that selection methods that use more features are more stable in their quality.

For some of the groups classified, a large number of the original features are essential to reach maximal performance, while in other cases, only a few features are sufficient for good separability. For example, as observed above, very few features are required to separate the porin group (only 3 features).

Finally, we are unable to find any specific subset of features that consistently dominates the entire set; the chosen ones range from AAC (e.g., in vacuole proteins) and AAG (the nucleosome group) to others, but only rarely includes the PTMs. The last observation seems to indicate that these signatures do not predict functional protein groupings, perhaps since identical modifications are often performed on differently functioning proteins [29]. The biophysical and enrichment features (25 features) are also rarely selected by the FF or BE protocols.

3.4 Global vs. Local Features

As can be discerned from Fig. 4 (top), a representation of proteins using global features compares to local comparison-based features (SW), as classification using the global features (all or partial) yields superior results in 6 of the 10 groups. Also shown is that classification using only a subset of features, as obtained by the BE and FF methods, yields good results.

The quality in classification performance using global feature representations varies across the different groups tested. Some protein groups failed to classify with high precision (e.g., tricarboxylic acid cycle), while in other groups a very small set of features was found sufficient (e.g., porin activity). Nonetheless, using all 70 global features provided a very successful classification for all groups.

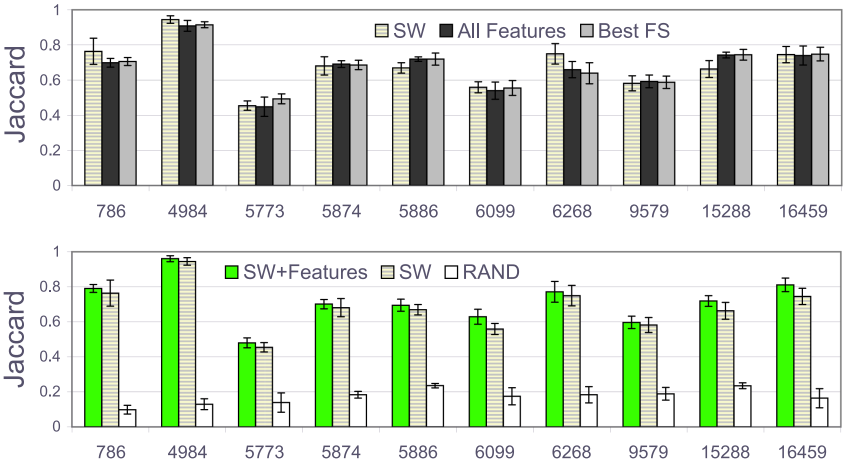


Fig. 4. Top: SVM results for the protein groups: local sequence similarities (SW: stripes), all 70 global features (All Features: black) and the best feature selection scheme (Best FS: gray). **Bottom:** Combination of both representations (SW + features: green), local sequence similarities (SW: stripes) and a random classification (RAND: white).

3.5 Combining Local with Global Features

Since both feature sets (SW and global) were transformed and scaled to a common representation (see Methods), it is possible to combine them into a unified dataset. This was performed in the following way: assuming that the N proteins are described by M global features, then the feature dataset matrix is *[NxM]* and the SW one is *[NxN]*. Combining the matrices is simply performed by appending them, resulting in a *[Nx(M+N)]* matrix.

Fig. 4 (bottom) demonstrates that naive combination of global and local features significantly improves the classification quality, compared to relying on either of them alone (paired t-test < 0.001, and < 0.05, respectively). This suggests that the two representations contain complementary information. Thus it would seem that combining these features is an effective practice and should be adopted for large-scale functional protein classification.

4 Discussion

In this study we show that characterization of protein families can be obtained by relying on a small set of global features that, in some cases, can be further reduced. In previous studies, when much richer feature sets were used [11, 12], the comparison with local features (SW) showed lower success rates. We hypothesize that the high-quality results described here are due to the small number of features that describe the data. This small size may facilitate the training and predictive capabilities of the classifier and, as a result, improves the classification.

We attempted to determine which global features and feature selection algorithms perform best in the task of protein function prediction. There is no one feature set that performed this task equally well for all groups, since only some groups seem ”easy” to predict in that they require few features to characterize them well. Nevertheless, when a given group was found to be ”easy”, then it was usually discovered by the FF method (or by using one of the predefined classes of features). On the other hand, single-wise feature selection (GREEDY) was prone to over-fitting and inferior to methods that consider the interplay between features and attempt to separate the training set in a holistic fashion (FF and BE). Therefore, it would seem wise to avoid such greedy methods that independently select features.

In summary, we have observed that the use of global sequence features compares with the use of local features in functional protein classification. Since the calculation of such global features is much faster (theoretically and in practice) than computation of local sequence alignments for all pairs of proteins to be compared, in future work we plan to assess the protein function classification problem using global features on a much larger scale (from the GO resource). In addition, since we have also shown that the combination of local and global sequence features succeed more than either method alone, it is certainly worthwhile for large-scale prediction algorithms to incorporate both protein representations. For computationally heavier methods that already use local sequence information (local alignment algorithms), the assimilation of global sequence properties as described here could be done at minimal overhead, yielding stronger prediction algorithms with little or no increase in computing time.

The scheme presented here was also applied to protein sets of major biological importance and to a 10-fold larger set (not shown). Success in separating kinases (the serine-threonine, tyrosine and uncharacterized), as well as nuclear proteins of the DNA from RNA biosynthesis proteins, suggest that, at the coarse level of classification, protein groups may be characterized by a very minimal set of global features. On the other hand, substantial improvement was achieved for proteins that often fail by sequence similarity, such as snake toxins and cytokines.

References

[1] Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Res. 25(17), 3389–3402 (1997)

[2] Scheeff, E.D., Bourne, P.E.: Application of protein structure alignments to iterated hidden markov model protocols for structure prediction. BMC Bioinformatics 7, 410 (2006)

[3] Portugaly, E., Harel, A., Linial, N., Linial, M.: Everest: automatic identification and classification of protein domains in all protein sequences. BMC Bioinformatics 7, 277 (2006)

[4] Gribskov, M., McLachlan, A.D., Eisenberg, D.: Profile analysis: detection of distantly related proteins. PNAS 84(13), 4355–4358 (1987)

[5] Yona, G., Levitt, M.: Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. J. Mol. Biol. 315(5), 1257–1275 (2002)

[6] Levitt, M., Gerstein, M.: A unified statistical framework for sequence comparison and structure comparison. PNAS 95(11), 5913–5920 (1998)

[7] Rost, B.: Topits: threading one-dimensional predictions into three-dimensional structures. Proc. Int. Conf. Intell. Syst. Mol. Biol. 3, 314–321 (1995)

[8] Frith, M.C., et al.: The abundance of short proteins in the mammalian proteome. PLoS Genet 2(4), e52 (2006)

[9] Friedberg, I., Kaplan, T., Margalit, H.: Glimmers in the midnight zone: characterization of aligned identical residues in sequence-dissimilar proteins sharing a common fold. Proc. Int. Conf. Intell. Syst. Mol. Biol. 8, 162–170 (2000)

[10] Wu, C.H., Apweiler, R., Bairoch, A., Natale, D.A., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Mazumder, R., O'Donovan, C., Redaschi, N., Suzek, B.: The universal protein resource (uniprot): an expanding universe of protein information. Nucleic Acids Res. 34(Database issue), D187–D191 (2006)

[11] Kunik, V., Solan, Z., Edelman, S., Ruppin, E., Horn, D.: Motif Extraction and Protein Classification. In: IEEE Computational Systems Bioinformatics Conference (CSB'05), pp. 80–85. IEEE Computer Society Press, Los Alamitos (2005)

[12] Cai, C.Z., Han, L.Y., Ji, Z.L., Chen, X., Chen, Y.Z.: Svm-prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Res. 31(13), 3692–3697 (2003)

[13] Bateman, A., Birney, E., Cerruti, L., Durbin, R., Etwiller, L., Eddy, S.R., Griffiths-Jones, S., Howe, K.L., Marshall, M., Sonnhammer, E.L.: The pfam protein families database. Nucleic Acids Res. 30(1), 276–280 (2002)

[14] Syed, U., Yona, G.: Using a mixture of probabilistic decision trees for direct prediction of protein function. In: Proceedings of RECOMB, pp. 224–234 (2003)

[15] Chou, K.C.: Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics 21(1), 10–19 (2005)

[16] Kahsay, R.Y., Gao, G., Liao, L.: An improved hidden markov model for transmembrane protein detection and topology prediction and its applications to complete genomes. Bioinformatics 21(9), 1853–1858 (2005)

[17] Chou, K.C., Cai, Y.D.: Predicting protein quaternary structure by pseudo amino acid composition. Proteins 53(2), 282–289 (2003)

[18] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)

[19] Camon, E., Barrell, D., Lee, V., Dimmer, E., Apweiler, R.: The gene ontology annotation (goa) database–an integrated resource of go annotations to the uniprot knowledgebase. In Silico Biol. 4(1), 5–6 (2004)

[20] Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. 147(1), 195–197 (1981)

[21] Hulo, N., et al.: The prosite database. Nucleic Acids Res. 34(Database issue), D227–D230 (2006)

[22] Gasteiger, E., Gattiker, A., Hoogland, C., Ivanyi, I., Appel, R.D., Bairoch, A.: Expasy: The proteomics server for in-depth protein knowledge and analysis. Nucleic Acids Res. 31(13), 3784–3788 (2003)

[23] Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 292(2), 195–202 (1999)

[24] Eichacker, L.A., Granvogl, B., Mirus, O., Muller, B.C., Miess, C., Schleiff, E.: Hiding behind hydrophobicity. transmembrane segments in mass spectrometry. J. Biol. Chem. 279(49), 50915–50922 (2004)

[25] Skoufos, E.: Conserved sequence motifs of olfactory receptor-like proteins may participate in upstream and downstream signal transduction. Receptors Channels 6(5), 401–413 (1999)

[26] Henikoff, J.G., et al.: Increased coverage of protein families with the blocks database servers. Nucl. Acids Res. 28(1), 228–230 (2000)

[27] Conticello, S.G., Pilpel, Y., Glusman, G., Fainzilber, M.: Position-specific codon conservation in hypervariable gene families. Trends Genet 16(2), 57–59 (2000)

[28] Paulsen, I.T., Park, J.H., Choi, P.S., Saier, M.H.: A family of gram-negative bacterial outer membrane factors that function in the export of proteins, carbohydrates, drugs and heavy metals from gram-negative bacteria. FEMS Microbiology Letters 156(1), 1–8 (1997)

[29] Chakrabarti, S., Lanczycki, C.J.: Analysis and prediction of functionally important sites in proteins. Protein Sci. 16(1), 4–13 (2007)

Epilogue

In this section we aim to provide several comprehensive statements, rather than provide summaries and conclusions. The latter are provided at the end of each chapter. We wish to emphasize six basic truths that emerge from our analysis.

- *"When possible, let the data speak for themselves"*

A basic motivation of our research was to extract information, and thus infer significant biological knowledge, merely by observing the data. We interpret this motivation by preferring unsupervised methods when possible. While the statement above may sound quite simplistic and it certainly cannot be applied to all scientific fields, we find that the special characteristics of genomic data dictate such an approach, at least in the exploratory stage. The rationale behind adopting the unsupervised strategy was: (1) lack of agreed labeled data, (2) over-fitting, sampling bias and other failings that are results of the train-test splitting in supervised methods and (3) the hope to allow for the emergence of more reliable and sometimes surprising results by acting in an unbiased manner.

Following this principle we focused on developing unsupervised algorithms (chapter 2-3), and provided several data-driven criteria for evaluating those algorithms. Examples for the latter are (1) in UFF, the pattern of CE scores over all features, may testify how well the method fits a given dataset (see Appendix A), (2) the UFF method may also serve as an internal test-bed for comparing between several imputation methods (chapter 2.1), (3) in hierarchical clustering, the structure of the tree may reflect the number of clusters and the algorithm-data fit (chapter 3), and (4) the algorithms optimizer as a data-driven comparison framework for clustering (chapter 4.1).

- *"When possible, let mathematics help interpreting biology"*

The algorithms we suggest are based solely on mathematical and statistical foundations, ignoring any specific biological considerations. Following this principle, we were able to obtain less biased results while providing generic, large-scale compatible algorithms. However, as the research was originally motivated by biological questions, all algorithms were applied to biological data. Furthermore, biology was a principal focal point of the inference part of each study. These mathematical driven algorithms led to some intriguing biological observations, such as relevant genes selected by UFF (chapter 2), surprising protein groups suggested by TDQC algorithm (chapter 3) and unexpected global features characterizing families of proteins (chapter 5).

- *“When mining genomic and proteomic data, don’t expect a ‘one-size-fits-all’ method”*

Throughout the research we realized, that the diversity of the data, ‘polluted’ by numerous intervening factors, makes it impossible to provide a single overall solution that best handles all cases. As a result, for every particular case a different algorithm and configuration may be preferred. Therefore, we put large emphasis on developing appropriate evaluation methods for comparing between imputations, filtering and clustering algorithms.

- *“Do not ignore less explored directions”*

Most of the directions we explored have not been well studied in the literature, particularly in computational biology. For example, there are only a handful of global (Top Down) hierarchical clustering algorithms or unsupervised feature filtering methods. Furthermore, these methods are rarely applied to experimental data. This research suggests that currently overlooked approaches should not be neglected. Surprisingly, according to our experience, these methods are shown to be very effective when exploring genomic and proteomic data.

- *“Any model must be backed up by praxis”*

Our research was guided by realistic and applicative motivations, not limited only to theoretic perspectives. As a result, all our algorithms were applied to experimental datasets. Always, a software tool was developed for the corresponding algorithm. For instance, the COMPACT package, which has been made freely available to academic usage, has been accessed to date more than 5,500 times, downloaded more than 750 times and served as the basis for two graduate courses. As users of these tools may be biologists or medical researchers who are not data mining experts, providing intuitive, graphical and user-friendly applications is of prime interest.

- *“When possible, follow the Occam’s Razor principle”*

Last but not least, throughout the research, we were motivated to follow the law of parsimony. Hence, we favored solutions that are easy to comprehend and fast to implement. Additionally, a main focus of our research was to find a minimalist set of features or parameters that describe hidden patterns in the data.

Chapter 7

Appendix: Unsupervised Feature Filtering and Instance Selection

Assaf Gottlieb[*†‡] Roy Varshavsky[†§]
Michal Linial[**] David Horn[‡]

Abstract

Feature selection is an important preprocessing task in the analysis of complex data. Selecting an appropriate subset of features can improve classification or clustering and lead to better understanding of the data. An important example is that of finding an informative group of genes out of thousands that appear in gene-expression analysis. Numerous supervised methods have been suggested but only a few unsupervised ones exist.

We present an Unsupervised Feature Filtering (UFF) approach, based on estimating the contribution of each feature to the Singular Value Decomposition (SVD) of the data. The estimate is based on SVD-entropy, thus taking into account the context of all other features. UFF ranks all features and provides a natural selection of the preferred group of features. We demonstrate that UFF outperforms other unsupervised selection methods, and analyze the statistical nature of its selected features. In addition, we propose criteria indicating which datasets are amenable to feature selection by UFF. Relying on a formalism similar to UFF we propose also an Unsupervised Instance Selection (UIS) method. UIS allows selection of instances whose characteristics deviate from all others. The latter may be disregarded at the clustering stage. Our methods are demonstrated and tested on known benchmarks.

Supplementary Material: http://adios.tau.ac.il/UFF

1. Introduction

The present information age is characterized by exponentially increasing data, e.g. in documents, records of various kinds or biological data. Improved experimental techniques, such as high throughput methods in biology, allow for the measurement of thousands of features (genes) for each instance (single gene-expression microarray per patient). This leads to a flood of data, whose analysis calls for preprocessing in order to reduce noise and enhance the signal through dimensionality reduction. This is important for both enabling the application of various categorization techniques and allowing for biological inference from the data.

Dimensionality reduction algorithms are usually categorized as *extraction* or *selection* methods. In feature extraction, all features are transformed into a lower dimension space, while in feature selection, a subset of the original features is selected. A benefit of the latter is the ability to attach meaning to the selected features. This is important both for exploration of the biological reality and for preparing a more concise experimental layout. The methods to be studied here are categorized as feature selection.

It is customary to divide feature selection methods into two types: *supervised*, in which a target function is known and one tries to rank features or optimize some objective function relative to it, and *unsupervised*, in which one has no information regarding the instances. In

[*] To whom correspondence should be addressed

[†] These authors contributed equally

[‡] School of Physics and Astronomy, Tel Aviv University

[§] School of Computer Science and Engineering, The Hebrew University of Jerusalem

[**] Department of Biological Chemistry, Institute of Life Sciences, The Hebrew University of Jerusalem

practice, due to the abundance of data, most of it unlabelled, it seems that most problems call for an unsupervised approach.

While supervised feature selection methods are abundant, unsupervised methods are scarce, most of them tested on labeled data [9]. Nevertheless, unsupervised feature selection methods may play an important role even in supervised cases. Being unbiased by the labeling of the instances, unsupervised feature selection can be used as a preprocessing tool for supervised learning algorithms providing reduction of overfitting (for a comprehensive review we refer to [9]). As described in [5], feature selection from unsupervised data can be applied at three different stages: before, during and after clustering. Methods that operate before clustering are referred to as *filter* methods. Common methods of unsupervised feature filtering rank features according to either *(1)* their projection on the first principal components [25] , *(2)* their normalized range,*(3)* entropy or *(4)* variance of the feature as calculated from its values on all instances [9] [13]. All these methods estimate the importance of each feature independently of all others.

Our Unsupervised Feature Filtering (UFF) algorithm [23] differs from aforementioned methods in that it ranks features based on a criterion that involves all other features. It also provides a natural cutoff for selecting the number of features. Our aim in this article is to suggest UFF as a strong preprocessing tool by *(1)* exploring the properties of UFF and the features it selects, *(2)* suggesting indicators for the ability to apply the method to certain datasets and *(3)* extending it by proposing a method called Unsupervised Instance Selection (UIS) for inspecting and eliminating potential outlier instances.

The outline of the article is as follows: in the next sections we introduce the concept of UFF (in section 2.1), explore the properties of UFF using example datasets (2.2), compare UFF with other filtering methods (2.3), analyze which datasets can be evaluated by the UFF method (2.4). Finally we describe the UIS method in section 3, and discuss some aspects of our findings in section 4.

2. Unsupervised Feature Filtering (UFF)

2.1 Selecting Features

In many problems, such as gene expression, all features are of similar nature, yet only some of them bear relevance to the data under investigation. Looking for the relevant features is the goal of feature selection. The main idea of our approach is to eliminate one feature at a time from the data matrix in order to estimate the effect of this feature on the data. In practice we use the Singular Value Decomposition (SVD) procedure. Let *A* denote a matrix, whose elements A_{ij} denote the measurement of feature *i* on instance *j*, e.g. expression of gene *i* under condition *j*. SVD decomposes the original matrix *A* into $A=USV^T$, where *U* and *V* are unitary matrices whose columns form orthonormal bases. The diagonal matrix *S* is composed of singular values (s_k) ordered from highest to lowest. SVD is a common technique for dimensionality reduction. Conventionally, it is either used in feature extraction by truncating *S* using only the first *r* singular values, which results in the best *r*-rank approximation of the original matrix in the least-square sense, or by exploring the *r* leading eigenvectors [24] [1]. UFF uses the information contained in the singular values differently, in order to select the features.

Let *q* be the rank of the matrix (*q≤min(n,m)*, where *n* is the number of instances and *m* is the number of features). Using the singular values, s_k, one may define the normalized relative squared values ρ_k [24] [1]:

$$\rho_k = s_k^2 \Big/ \sum_{i=1}^{q} s_i^2 \qquad (1)$$

A dataset that is characterized by only a few high normalized singular values, whereas the rest are significantly smaller, reflects large redundancy in the data. On the other hand, non-redundant datasets lead to uniformity in the singular values spectrum. UFF exploits the property of the spectrum in order to measure how each feature *i* influences this redundancy, while favoring features which decrease redundancy. The score of a feature *i* is defined using a leave-one-out principle. A function *f* is calculated on the set of all singular values for the original matrix and for the corresponding set of the matrix without feature *i*. The difference in the values of *f* defines the score of each feature *i*. In this work, we use the SVD-entropy (*H*) as the function *f* [1] [4] (note that this 'Shannon'-like function does not use probabilities). The score of a feature can be thus regarded as its contribution to the SVD-entropy.

$$f = H = -\frac{1}{\log(q)} \sum_{k=1}^{q} \rho_k \log(\rho_k) \qquad (2)$$

Other functions may be used instead of *H*. They have to be monotonic and vary from a maximum, when all singular values are equal, to a minimum when there is only one singular value bigger than zero. Two such functions that we tested are the negative value of sum of squares and the geometric mean (expressions 3 and 4, respectively). The results using these functions are very similar to those obtained using the SVD-entropy, hence we will not elaborate further on them.

$$f_{ss} = -\sum_k \rho_k^{\,2} \qquad (3)$$

$$f_{GM} = \left(\prod_{k} \rho_k \right)^{1/q} \quad (4)$$

Figure 1 displays the results after applying the UFF algorithm to two different datasets (see section 2.3), and sorting the features according to the decreasing score of the UFF. Clearly, one can divide the features into three groups:

1. Features with positive score. These features increase the entropy.
2. Neutral features. These features have negligible influence on the entropy.
3. Negative score features. These features decrease the entropy.

Note that a majority of all features falls into group 2, while groups 1 and 3 represent minorities. We argue that the most relevant features belong to group 1. The rationale behind picking the positive score features is that, because they increase the entropy, they decrease redundancy. Hence we may expect that instances may be better separated in the space spanned by these features. Further analysis of this group and its comparison with the two other groups is presented in section 2.2.

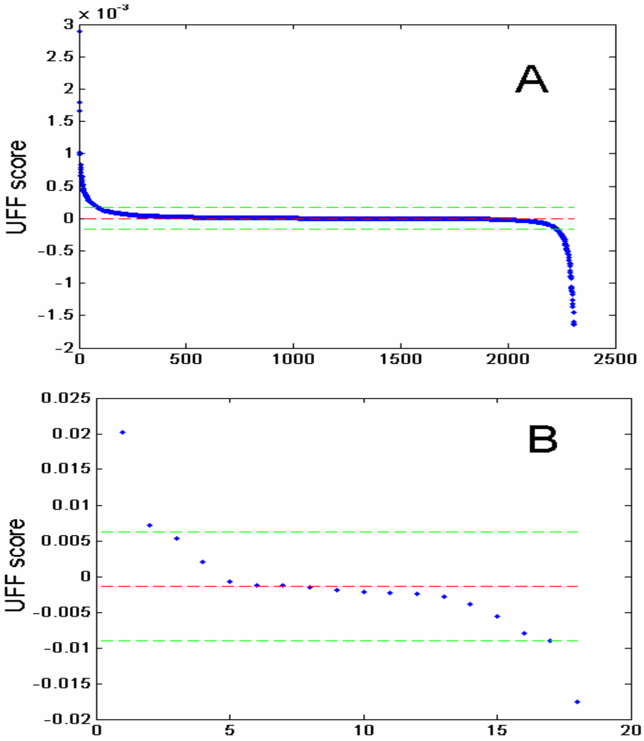


Figure 1. UFF Scores of the (A) 2308 genes of SRBCT and (B) 18 virus features, ordered by decreasing scores. Dashed lines represent mean(score)±std(score). Note that the two different datasets have similar characteristics. Irrespective of the number of features and the values of their scores, we find clear separation into three groups of features.

Features whose scores lie above the mean+std are selected as the relevant ones. The division supplied by the std defines a set of m_c selected features (m_c<m). We refer to this selection as Simple Ranking (SR). Two alternative selection methods are Forward Selection and Backward Elimination [23]. Here we will concentrate on SR. Results of the alternative UFF methods appear in the supplementary material.

2.2 Properties of Selected Features

We investigate the features selected by UFF, by looking at their statistical properties. First we plot in Figure 2 the mean (A) and variance (B) of all features (as measured on all instances). These are shown for the SRBCT dataset used in Figure 1A. Most features belonging to the second (neutral) group possess low mean and variance. It is evident that both the positive score features and the negative score features have high mean and variance. This explains a major difference between UFF and the Variance Selection method: while UFF selects features from group 1, Variance Selection chooses features from both groups 1 and 3. In this context, it is noteworthy that datasets of this nature (such as of gene-expression) should not undergo any zero-mean normalization, as the averages of the various feature bares meaningful information.

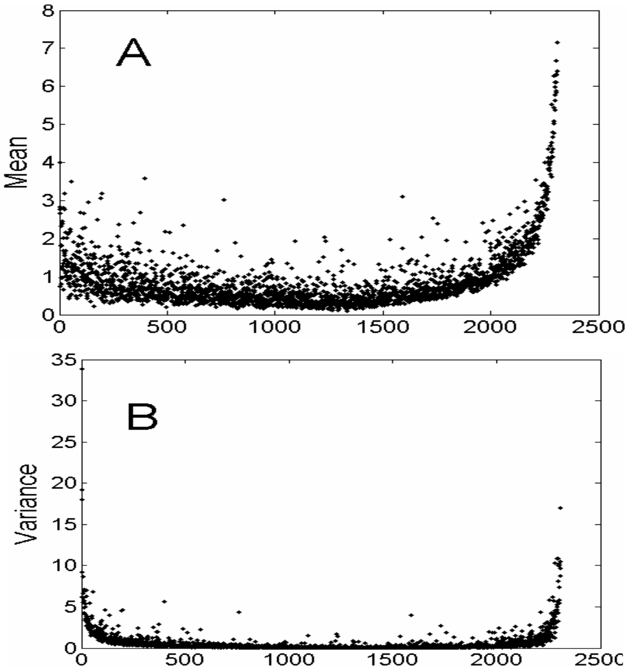


Figure 2. (A) mean and (B) variance of the SRBCT dataset (X axis refers to genes ordered according to UFF score).

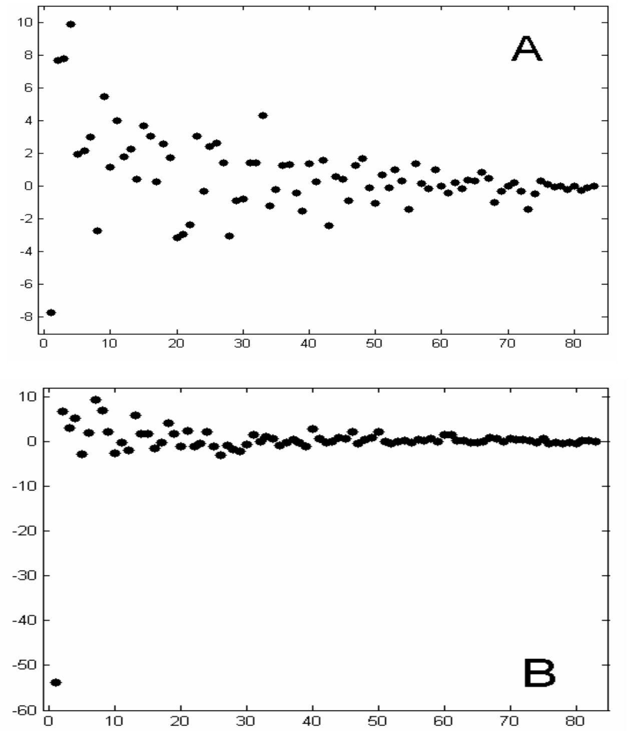


Figure 3. Projection on the 83 principal components of a typical - (A) positive score (B) negative score - feature from the SRBCT dataset. Note the outstanding value of PC1 in B.

An important difference between the positive (group 1) and negative (group 3) features is displayed in Figure 3. This figure shows the projection of typical positive and negative features (A and B, respectively) on the SVD eigenvectors (or principal components, PCs) of the original data matrix. Positive score features have relatively evenly distributed projections on the PCs, while negative score features project strongly on the first. It is the latter property that explains the negative score: by preferring the leading principal component these features decrease SVD-entropy. We present in the Appendix a proof that when a feature lies only on the first PC, it is bound to have a negative score. The proof for the SVD-entropy function can be extended to cover also the alternative measures of equations 3 and 4.

The differences in projection on the principal components between the positive and negative scored features, may provide an explanation for the difference between our approach and the sparse-PCA approach that have recently been suggested [25]. The latter selects features that in essence, correlate mainly with the first leading principal components, while UFF prefers features that tend to distribute evenly along most of the principal components.

Furthermore, we also find that the negative score features have skewness close to zero and kurtosis close to three. Hence we conclude that group 3 features, discarded by UFF but selected by Variance Selection, possess wide Gaussian distributions. This means that Variance Selection contains noisy features and explains their inferior results demonstrated in the next section.

2.3 Data and Results

In order to demonstrate the performance of UFF, and to compare it with other feature selection methods, we apply it to two representative datasets. The first is the small round blue cell tumor (SRBCT) gene-expression dataset that was first introduced in [16], and includes cDNA microarray measurements of 2308 genes (features) for 83 patients (instances). The instances are categorized into four types of tumors: Burkitt lymphoma, Ewing sarcoma, Neuroblastoma and Rhabdomyosarcoma. The second dataset, originally described by [6] and analyzed more thoroughly by [19], contains 61 rod-shaped viruses affecting various crops. There are 18 measurements of Amino Acid Compositions (AAC) for the coat proteins of the virus serving as 18 features. The viruses are classified into four classes: Hordeviruses, Tobraviruses, Tobamoviruses and Furoviruses. It is worth mentioning, that neither the UFF nor the other algorithms use these labels either at the filtering or at the clustering stage.

In order to assess the quality of the filtering methods, clustering of the instances is performed on the filtered dataset. In the cases described below, clustering is based on the QC algorithm [11] (it is shown in [23] and in the supplementary material that similar conclusions regarding feature filtering are obtained when applying other clustering algorithms, e.g., hierarchical clustering and *K*-Means). Assessment of clustering quality with respect to expert classification of the data is measured using the popular criterion of Jaccard score (*J*) [10, 14, 20]

Figures 4 and 5 display the clustering results for the SRBCT and viruses datasets, respectively, when several unsupervised filtering methods are applied. The methods compared are UFF, normalized range (range values of the feature normalized by the minimal value), Variance and Entropy (of feature values over the instances), and random selection (for each number of features we use 50 repeats of random selections from the total set of features). The dashed line denotes the score obtained when using all features.

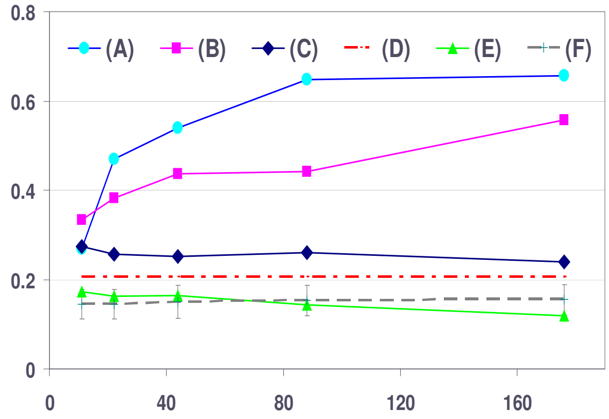


Figure 4. Clustering results (Jaccard score, Y axis) of the SRBCT dataset as a function of the number of features selected by each method: (A) UFF, (B)Normalized range, (C) Variance, (D) All, (E) Feature Entropy and (F) Random.

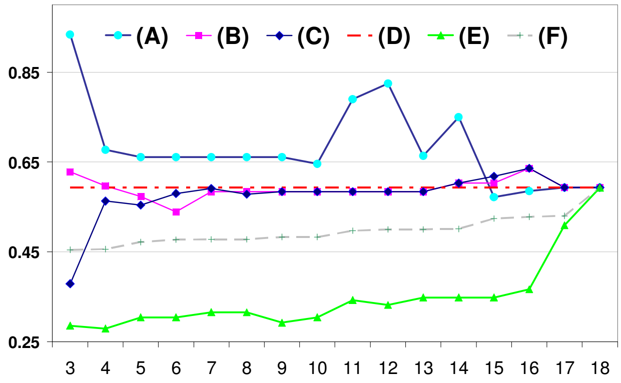


Figure 5. Clustering results (Jaccard score, Y axis) of the viruses dataset as a function of the number of features selected by each method: (A) UFF, (B) Normalized range, (C) Variance, (D) All, (E) Feature Entropy and (F) Random

The two figures show that UFF outperforms other methods, especially when the selected group of features is relatively small. Whereas the results are shown as function of the number of features that are retained, note that UFF contains an estimate (m_c) of the number of features to be selected. These values are 88 for the SBRCT dataset (Figure 4) and 3 for the virus data (Figure 5). At both values we witness the largest difference in clustering quality between UFF and the other methods.

2.4 When is UFF Applicable

While UFF works very well on many datasets, including most gene-expression data, we have found datasets where selection according to UFF is not effective. Figure 6 presents two such examples: datasets of stocks [21] and cell-cycle gene-expression [22]. On both, UFF did not lead to improved clustering (not shown). We note that the distributions in Figures 6 and 7 are somewhat different from Figure 1. In particular, group 2 features display large variance among their scores.

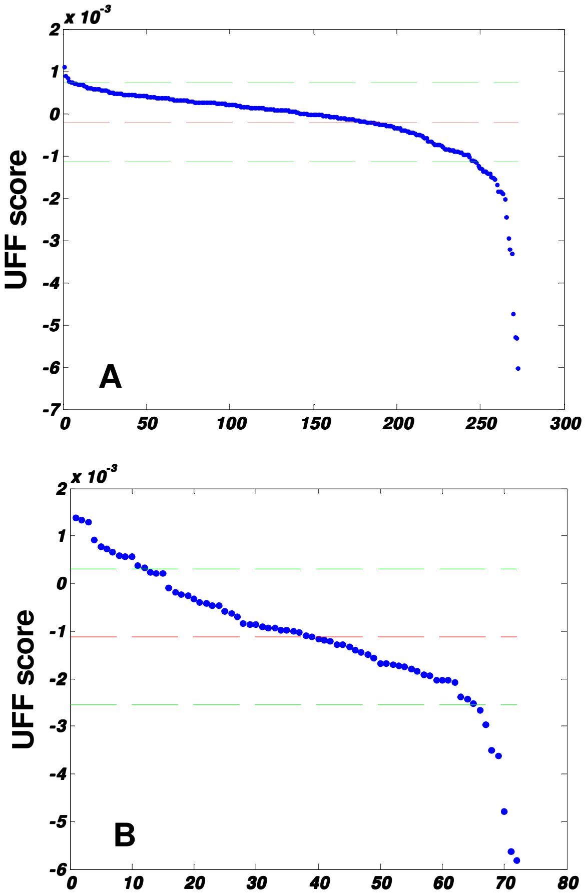


Figure 6. UFF Scores of (A) stocks dataset and (B) cell-cycle gene-expression dataset, ordered by decreasing UFF score.

Working with more than a dozen datasets from different domains, all shown in the supplementary material, we have found measures that allow for separation between 7 datasets on which UFF is effective from 5 datasets in which it is not. One such measure is the normalized entropy of the squares of UFF scores. This, as well as another measure, is presented in the Appendix. They allow for a prior estimate on whether UFF should be employed.

3. Unsupervised Instance Selection (UIS)

The data-matrix A contains information on instances in terms of features and features in terms of instances, and the singular values are common to both. One may therefore consider a 'leave-one-out' measure applied to instances. This is the Unsupervised Instance Selection (UIS) method, to be studied here. It turns out to be useful for identifying outliers among the instances that may be removed in order to provide a more homogeneous dataset.

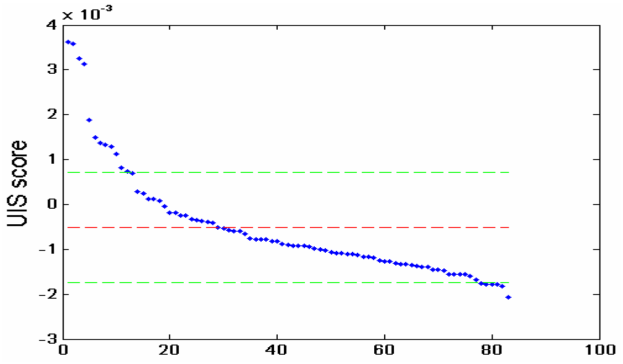


Figure 7. UIS scores of the instances of the SRBCT dataset, ordered by decreasing score

Evaluation of the UIS method is done in the same manner as for UFF, i.e. clustering the remaining instances and comparing against expert labeling. Table 1 displays clustering results for the SRBCT and viruses datasets for (a) all features and instances, (b) the UIS-filtered instances while keeping all features, (c) the UFF features and all instances, and, (d) the joint selection of instances and features by UIS and UFF respectively. UIS followed by UFF markedly improves the clustering quality, having Jaccard scores of 0.88 and 0.95 for the two datasets, respectively. The results were compared to clustering done on the datasets in which instances were randomly removed (13 and 6 instances, in the SRBCT and viruses datasets respectively). No improvement of the Jaccard score was found. Hence we conclude that removal of UIS selected instances is indeed efficient. The UIS eliminated instances are found to be distributed homogenously among the four classes in both datasets. Other datasets appear in the supplementary material.

Table 1: Clustering quality (Jaccard scores) for SRBCT and viruses datasets using all the features, UIS, UFF and UFF+UIS

	All	UIS	UFF	UIS+UFF
SRBCT	0.21	0.20	0.65	**0.88**
Virus	0.59	0.68	0.93	**0.95**

4. Conclusions

We present and explore UFF, an unsupervised approach that scores and ranks each feature according to its influence on the singular values distribution. By applying a leave-one-out method, scoring of each feature is determined with regard to all other features, and not independently as by other standard methods.

A statistical characterization of the selected features shows that our method selects features of high variance (over instances), but only those that do not have large correlation with the first principal component. It turns out that thus we ignore noisy features that have Gaussian distributions.

By studying various empirical datasets and evaluating different scoring functions we show that our approach is generic, and can identify the subset of relevant features. In contradistinction to other methods we can estimate the size of the group of selected relevant features.

UFF is a heuristic method which exposes its strength in realistic application. Nevertheless, not all datasets are amenable to feature selection by UFF. We propose criteria for deciding when UFF application is effective.

We extend the capabilities of UFF by introducing the Unsupervised Instance Selection (UIS) method. Application of the latter followed by UFF fulfills three important goals: (1) identify and remove outliers from the dataset, (2) identify and select the most informative features and (3) improve the clustering quality.

Acknowledgments

We thank Alon Kaufman and Nati Linial for stimulating discussions and suggestions.

A.G. is supported by the Colton Foundation.

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 DIAMONDS consortium.

References

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

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

[3] D. G. Beer, S. L. R. Kardia, C.-C. Huang, T. J. Giordano, A. M. Levin, D. E. Misek, L. Lin, G. Chen, T. G. Gharib, D. G. Thomas, M. L. Lizyness, R. Kuick, S. Hayasaka, J. M. G. Taylor, M. D. Iannettoni, M. B. Orringer and S. Hanash, *Gene-expression profiles predict survival of patients with lung adenocarcinoma*, Nat Med, 8 (2002), pp. 816-824.

[4] P. A. Devijver and J. Kittler, *Pattern recognition : a statistical approach*, Prentice-Hall, Englewood Cliffs, N.J, 1982.

[5] J. G. Dy and C. E. Brodley, *Feature Selection for Unsupervised Learning*, J. Mach. Learn. Res., 5 (2004), pp. 845-889.

[6] C. Fauquet, D. Desbois, D. Fargette and G. Vidal, *Classification of furoviruses based on the amino acid composition of their coat proteins*, Dev. Appl. Biol, 2 (1988), pp. 19-36.

[7] R. Feynman, P, *Forces in Molecules,* Physical Review, 56 (1939), pp. 340 - 343

[8] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield and

E. S. Lander, *Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring*, Science, 286 (1999), pp. 531-537.

[9] I. Guyon and A. Elisseeff, *An Introduction to Variable and Feature Selection*, Journal of Machine Learning Research, 3 (2003), pp. 1157--1182.

[10] J. Handl, J. Knowles and D. B. Kell, *Computational cluster validation in post-genomic data analysis*, Bioinformatics, 21 (2005), pp. 3201-3212.

[11] Hellman-Feynman, *Feynman-Hellman theorem of quantum mechanical forces was originally proven by P. Ehrenfest, Z. Phys. 45, 455 (1927), and later discussed by Hellman (1937) and independently rediscovered by Feynman (1939).*

[12] H. Hellman, *Einfuhrung in die Quantenchemie*, Deuticke, Leipzig and Vienna, 1937.

[13] J. Herrero, R. Diaz-Uriarte and J. Dopazo, *Gene expression data preprocessing*, Bioinformatics, 19 (2003), pp. 655-656.

[14] P. Jaccard, *Nouvelles recherches sur la distribution florale*, Bulletin de la Societé Vaudoise des Sciences Naturelles, 44 (1908), pp. 223-270.

[15] A. Kagian, G. Dror, T. Levyand, D. Cohen-Or and E. Ruppin, *A humanlike predictor of facial attractiveness*, *Neural Information Processing Systems (NIPS)*, 2006.

[16] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson and P. S. Meltzer, *Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks*, Nat Med, 7 (2001), pp. 673-679.

[17] W. Li, P. Goovaerts and M. Meurens, *Quantitative Analysis of Individual Sugars and Acids in Orange Juices by Near-Infrared Spectroscopy of Dry Extract*, J. Agric. Food Chem., 44 (1996), pp. 2252-2259.

[18] S. O'Rourke and I. Herskowitz, *Unique and redundant roles for Hog MAPK pathway components as revealed by whole-genome expression analysis.*, Mol Biol Cell, 15 (2004), pp. 532-42.

[19] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996.

[20] R. Sharan and R. Shamir, *CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis*, ISMB 2000 (2000).

[21] N. Slonim, G. S. Atwal, G. Tkacik and W. Bialek, *Information-based clustering*, PNAS, 102 (2005), pp. 18297-18302.

[22] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein and B. Futcher, *Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization,* Mol. Biol. Cell, 9 (1998), pp. 3273-3297.

[23] R. Varshavsky, A. Gottlieb, M. Linial and D. Horn, *Novel Unsupervised Feature Filtering of Biological Data*, Bioinformatics, 22 (2006), pp. e507-513.

[24] M. Wall, A. Rechtsteiner and L. Rocha, *Singular Value Decomposition and Principal Component Analysis*, in D. Berrar, W. Dubitzky and M. Granzow, eds., *A Practical Approach to Microarray Data Analysis*, Kluwer, 2003, pp. 91-109.

[25] H. Zou, T. Hastie and R. Tibshirani, *Sparse Principal Component Analysis*, JCGS 15 (2006), pp. 262-286.

5. Appendix

5.1 Negative Score Feature Proof

One can prove that in the extreme case, where a feature is lying only on the first PC, it is bound to have a negative score. We shall now prove it for the SVD-entropy function. This proof can be extended to cover also the alternative measures in Equations 3 and 4.

Starting with the positive-definite correlation matrix C, defined as

$$C = A^T A = VS^2V^T \tag{5}$$

for the data matrix A of M features by N instances (where, without loss of generality we assume $N \le M$), we use its eigenvalues to define:

$$c_i = s_i^2,\ \rho_i = \frac{c_i}{T},\ T = \sum_{j=1}^{N} c_j,\ K = -\sum_{j=1}^{N} c_j \log(c_j \tag{6}$$

T is positive definite. SVD entropy can be related to K through

$$S = -\sum_{i=1}^{N} \rho_i \log(\rho_i) = \frac{K}{T} + \log\left(T\right) \tag{7}$$

where, for simplicity, we dropped the normalization constant (log(N)) in the definition of S

Consider the small perturbation of adding one feature to the matrix A. The assumption of a small perturbation generally holds for a large enough number of features. Using equation (7), we can write the resulting change of S as

$$TdS = dK + (1 - \frac{K}{T})dT \tag{8}$$

If an added feature projects only on the first PC, it can change only the first singular value. It follows then that

$$dT = dc_1,\ dK = \text{-}dc_1(1 + \log\left(c_1\right)) \tag{9}$$

Plugging the terms in (9) into equation (8), we arrive at

$$TdS = \frac{TdK + (T-K)dT}{T} = -\frac{dc_1}{T}(K + T\log\left(c_1\right)) < 0 \tag{10}$$

which means that adding such a feature always leads to reduction of entropy.

To complete the proof we show that the right hand side is indeed negative. T is positive, and so is also the sum of the two terms in the bracket, since c_1 is the leading eigenvalue and the following inequality holds:

$$-K = \sum_{1}^{N} c_j \ln(c_j) < T \log(c_1) \qquad (11)$$

We now prove that dc_1>0. Note that, by definition,

$$dc_i = \sum_{m,n} V_{mi} C_{mn} V_{ni} \qquad (12)$$

The first order perturbation of the eigenvalues of *C* is related to the change of the original matrix *C* by the original unitary transformation *V*. This follows from the unitarity constraint on *V*

$$\sum_m dV_{mi} V_{mi} = 0 \qquad (13)$$

and is the discrete analog of the Hellman-Feynman theorem[11], [12], [7].

Adding a row to *A*, i.e. adding the feature vector f^{M+1} of size *N*, the correlation matrix *C* changes to

$$C_{mn} \rightarrow C_{mn} + f_n^{M+1} f_m^{M+1} \qquad (14)$$

Plugging it back into equation (12), we conclude the proof with showing the dc_1 is positive according to:

$$dc_i = \left(f^{M+1} \cdot V^i \right)^2 \qquad (15)$$

where V_i is the i.-th eigenvector of *C*.

Adjusting appropriately *S* and *K*, it is easy to prove this also for the sum of squares and the geometric mean functions mentioned in 2.1.

5.2 When is UFF applicable?

We present two measures that allow for a separation between datasets on which UFF is effective, from those in which it is not. The first is SE, an entropy-like measure on normalized squares of UFF score-values.

$$w_k = \frac{Score_k^2}{\sum_{i=1}^{M} Score_i^2} \qquad (16)$$

$$SE = -\frac{1}{\log(M)} \sum_{k=1}^{M} w_k \log(w_k) \qquad (17)$$

and the second is VE, an entropy-like measure on the variance-values (i.e. variance of feature-values on all instances).

$$z_k = \frac{Var(f_k)}{\sum_{i=1}^{M} Var(f_i)} \qquad (18)$$

$$VE = -\frac{1}{\log(M)} \sum_{k=1}^{M} z_k \log(z_k) \qquad (19)$$

Suitable datasets can then be defined as those lying below certain thresholds in both measures. We tested 7 'suitable' and 5 'not-suitable' datasets using UFF and clustering algorithms. VE seems to provide a better margin of separation between the two groups of datasets ('suitable' datasets' VE range between 0.6 and 0.86, whereas 'not-suitable' datasets' VE range between 0.96 and 0.98). The datasets' description appear in section 5.3 and UFF graphs are provided in the supplementary material.

5.3 Datasets

The numbers in curly brackets denote the number of features x the number of instances. The numbers in square brackets reference the references section.

1. Small-Round-Blue-Cell-Tumor (SRBCT) {2308x83} [16]
2. Leukemia dataset 1 {7129x72} [8]
3. Leukemia dataset 2 {12582x72} [2]
4. Yeast microarray {5827x133} [18]
5. Virus {18x61} [6]
6. Facial-slopes {3486x91} [15]
7. Lung cancer {4966x96} [3]
8. Stocks {273x487} [21]
9. Facial-distances {3486x91} [15]
10. Orange Juice {700x218} [17]
11. Cell Cycle {72x798} [22]
12. Movies {943x1682} [21]

תוכן העניינים

מחקרנו שערכנו הונחה ע״י מניעים יישומיים וריאליסטיים. לאור זאת, כל האלגוריתמים נוסו על נתונים מחקריים אמיתיים (ולא על בסיס סימולציה). בכל המקרים חבילת תוכנה פותחה וסופקה עם האלגוריתם. לדוגמה לחבילת COMPACT, המסופקת באופן חופשי לשימוש אקדמי באינטרנט, נרשמו מעל 6000 צפיות ומעל 800 הורדות. החבילה אף היוותה בסיס לשני קורסים מתקדמים באוניברסיטות שונות. זאת ועוד, מכיוון שמרבית המשתמשים הפוטנציאליים של תוכנות אלו עשויים להיות ביולוגים או חוקרים רפואיים, אשר אינם מומחי כריית מידע, חיוני היה לפתח כלים גראפיים ידידותיים למשתמש.

לסיום, בכל שלבי המחקר, ולאורך כל רבדיו שאפנו לציית לעקרון *הפרסימוניות של אוקהם* (Occam's razor principle). לכן העדפנו פתרונות פשוטים, קלים להסבר ולמימוש. בנוסף, דגש מיוחד ניתן לאיתור קבוצות מינימליות של תכוניות או פרמטרים, המתארים תבניות חבויות בנתונים.

קדם תהליך מפוקח (בחירת תכוניות)

בשונה מבחירת תכוניות לא מפוקחת, שיטות מפוקחות נלמדו באופן מעמיק בתחום, ויישומים של שיטות אלו נפוצים מאוד. דוגמאות נפוצות הינן 'הכנסה קדימה' (forward insertion), 'הוצאה אחורה' (backward elimination), בחירה בשלבים (stepwise selection) ודירוג לפי מדדים סטטיסטיים (לדוגמה *t*-test).

קטגוריזציה מפוקחת (סיווג)

אלגוריתמי סיווג לומדים תבניות לפי קבוצת אימון מתויגת, ומנסים להסיק כלל הכללה התקף לשאר הפריטים (קבוצת המבחן). בדומה לאשכול, Support Vector Machine (SVM), עצי החלטה ושיטות מפוקחות אחרות, נמצאו כיעילות בסיווג נתוני ביטוי גנים או נתונים רציפים.

במחקר, המשלב את שני שלבי הלמידה המפוקחת (בחירת תכוניות וסיווג), הנחנו, כי ניתן לסווג מחלקות פונקציונליות של חלבונים, המתוארים בעזרת קבוצה קטנה במיוחד של תכונות ביוכימיות (כגון משקל מולקולרי, הידרופוביות, נפיצות של חומצות אמינו). כדי לבחון הנחה זו, חלבונים, המתוארים על ידי אותן תכונות, סווגו בעזרת אלגוריתם SVM. בנוסף, שימוש באסטרטגיות בחירת תכוניות שונות, אפשר את מדידת התרומה של כל אחת מן התכוניות למשימת הסיווג. תוצאות המחקר הראו, כי קבוצה קטנה של תכוניות גלובליות, אשר במצבים מסוימים יכולה אף להצטמצם יותר, מספקת מידע מהותי לגבי שייכות החלבונים לקבוצות פונקציונליות ואחרות. זאת ועוד, מצאנו, כי שילוב של תכונות (מבוססות רצף) לוקליות וגלובליות משפר, באופן מובהק, את איכות הסיווג.

לסיום נציין, כי מספר מוטיבציות כלליות הניעו את המחקר. ראשית, ככל שניתן היה, אלגוריתמים לא מפוקחים הועדפו. הסיבות להעדפה זו בכריית מידע גנומי הינן: *(1)* בהיותם מוטים פחות על ידי גורמים לא רלבנטיים, אלגוריתמים אלו מאפשרים גילוי תופעות שאינן צפויות; *(2)* המידע הגנומי טרם מופה ותויג במלואו; *(3)* החלוקה לקבוצות אימון ומבחן בלמידה מפוקחת טומנת בחובה מספר בעיות (כגון התאמת יתר, הטיית המדגם).

האלגוריתמים שהצגנו מבוססים, באופן בלעדי, על עקרונות מתמטיים וסטטיסטיים, ומתעלמים לחלוטין משיקולים ביולוגים. כתוצאה מכך, השיטות הינן כלליות ולא מוגבלות לסוג נתונים ביולוגי כזה או אחר. מכיוון שהבנה ופרשנות ביולוגיים היו בסיס המחקר, כל האלגוריתמים נוסו ויושמו על נתונים גדולים אלו. דגש רב ניתן להפקת פרשנות ביולוגית בסוף כל מחקר.

בנוסף, כאשר עוסקים ביישומי כריית מידע, ובפרט במקרים של נתונים ביולוגים רבים ורועשים, אין זה סביר שגישה אחת תתאים לכל הבעיות. כלומר, לכל מקרה יש להתאים את הפתרון וקונפיגורציית הערכים המיטביים. בשל כך, מחקרנו שם דגש רב על פיתוח שיטות הערכה והשוואה בין אלגוריתמים וקונפיגורציות שונות.

מרבית הכיוונים אותם חקרנו אינם נפוצים בספרות המדעית. בפרט, שיטות בחירת תכוניות בלתי מפוקחות ואלגוריתמי אשכול היררכיים גלובליים, אשר כמעט ואינם מתוארים ומיושמים במחקרים. מחקר זה מציע לא להזניח גישות ושיטות שנהוג בדרך כלל, להתעלם מהן. באופן מפתיע, אותן שיטות נמצאו כיעילות ביותר, בחקירת מידע גנומי וחלבוני.

ביולוגית במחקרי סרטן; *(4)* מדד פנימי, המסופק עם האלגוריתם, מציע הערכה של מידת אפקטיביות השיטה, בהינתן הנתונים.; *(5)* השיטה הינה כללית דיה להיות מוכללת לסינון פריטים במקום תכוניות.

אשכול

אלגוריתמי אשכול מופעלים כדי למצוא קבוצות נבדלות, ובשאיפה, בעלי רלבנטיות. גישה מקובלת זו, נמצאה כיעילה ביותר בקיבוץ גנים או דוגמאות בניסויי ביטוי גני ובקיבוץ חלבונים ע״ס הדמיון הרצפי. שני תחומים, השייכים לאשכול, אשר נחקרו הינם שיקולים גלובליים באשכול, והערכת האשכול.

שיקולים גלובליים באשכול. כפי שפורט לעיל, אלגוריתמי אשכול מסוימים פותחו מלכתחילה על מנת להתמודד עם נתונים גנומיים ואילו אחרים, הינם יישומים סטנדרטים של שיטות למידה חישובית. אחד מן האלגוריתמים הסטנדרטים הנפוצים ביותר הוא האשכול ההירככי המגובב (agglomerative hierarchical). שיטת אשכול זו מיושמת ברובם המוחלט של המחקרים. מגבלה בולטת של אלגוריתם זה נעוצה בהתעלמותו משיקולים גלובליים בתהליך הגיבוב. לשם הטמעת שיקולים אלו באשכול, פיתחנו שני אלגוריתמים: *(1)* TDQC (Top-Down-Quantum-Clustering): גישה היררכית חדשנית הפועלת מלמעלה-למטה ומבוססת על צפיפות הפריטים במרחב. *(2)* גרסה גלובלית-לוקאלית (׳גלוקאלית׳) של אלגוריתם הגיבוב, המשקללת את כל יחסי הדמיון בין הפריטים. בחינה מקיפה הראתה, כי שני אלגוריתמים אלו מציגים ביצועים עדיפים על פני אלגוריתמים היררכיים אחרים. בחינת האלגוריתמים בוצעה על נתונים מתחומים שונים, הכוללים ביטוי גנומי, מסחר מניות, וקבוצות פונקציונאליות של חלבונים.

הערכת האשכול. נתונים מניסוים ביולוגיים בכלל, וכאלו המתקבלים משימוש בטכנולוגיות החדשניות בפרט, הינם רועשים ורבים מאוד (הן במספר הפריטים והן במספר התכוניות). התפיסה האנושית אינה מותאמת לבחינה של מידע כה גדול, רב מימדים ורועש. כתוצאה מכך, הערכה ויזואלית של תוצאות אשכול איננה ישימה. מכיוון שהשונות בנתונים הביולוגים כה רבה (שכן מדובר בתנאי ניסוי ומחקר שונים), אין זה סביר שאלגוריתם אשכול יחיד יימצא יעיל ועדיף על פני האחרים בכל מצב. יתרה מכך, מכיוון שאלגוריתמים רבים שונים באופיים, מציאת הפתרון המיטבי הינה משימה מאתגרת ביותר. כתוצאה מכך, פיתחנו שלוש תשתיות אלגוריתמיות יישומיות, המיועדות לסייע במשימות אלו: *(1) מטייב אלגוריתמי אשכול* (The Clustering Algorithms Optimizer) הינו קבוצת פרוצדורות לא מפוקחות, אשר סורקות את מרחב פתרונות האשכול, ובוחרות את הפיתרון המיטבי על סמך קריטריון פנימי, התלוי רק בנתונים. קריטריון זה מבוסס על קריטריון האינפורמציה הבייסיינית (Bayesian Information Criterion BIC). שיטה זו מתגברת על מגבלות שכיחות, כגון התבססות על פרמטרים חיצוניים (לדוגמה מספר האשכולות) או גורמים לא דטרמיניסטים; *(2)* *COMPACT* (Comparative Package for Clustering Assessment) מציג שיטה וקבוצת פרוצדורות המאפשרות השוואה ויזואלית וסטטיסטית של מספר אלגוריתמים וכן מכיל מדדים חיצונים (המבוססים על תיוג הפריטים) להערכת התוצאות; *(3) ClusTree* הינה חבילת תוכנה גראפית לאנליזה והשוואת אלגוריתמי אשכול היררכיים.

נהוג לחלק את שיטות כריית המידע לפי השלבים באנליזה בהם הם מופעלים (קדם-תהליכי או קטגוריזציה), ועל בסיס היותם מפוקחים (supervised) או בלתי מפוקחים (unsupervised). מכאן שניתן לסווג שיטות כריית מידע לארבע קבוצות עיקריות: *(1)* קדם-תהליך לא מפוקח (unsupervised preprocessing); *(2)* קטגוריזציה לא מפוקחת (אשכול); *(3)* קדם-תהליך מפוקח; *(4)* קטגוריזציה מפוקחת (סיווג). יש לציין, שחלוקה זו אינה בלעדית וגבולותיה מטושטשים, כך שקיימות שיטות המשלבות בין הקבוצות.

במחקר זה נלמדו ופותחו אלגוריתמים של כריית מידע מכל אחת מארבע הקבוצות לעיל, ובפרט ניתן דגש על שיטות בלתי מפוקחות. כן נעשו ניסיונות לפתח שיטות בתחומים אשר, יחסית, פחות נחקרו עד כה.

קדם-תהליך בלתי מפוקח

אחד השלבים הראשונים ברוב תהליכי האנליזה הינו קדם התהליך. בשלב זה, הנתונים מוכנים (לדוגמה ניקוי, נרמול, מילוי ערכים חסרים), 'רעש' מנוקה, פריטים (instances), או תכוניות שאינן רלבנטיות מוסרים, וממדיות מערך הנתונים מופחתת.

אחת הגישות הנפוצות לנרמול, הפחתת מימד וסינון רעשים בנתונים היא *מיצוי תכוניות* (feature extraction). אולם, ICA, PCA, SVD ושיטות מיצוי אחרות מעבירות את **כל** התכוניות למימד נמוך, ולכן לא מספקות פרשנות לתכוניות מסוימות, כפי ששיטות *בחירת תכוניות* (feature selection) מאפשרות.

מרבית החוקרים, בעיקר באופן שאינו מודע, מפעילים מספר שיטות בחירה בלתי מפוקחות (לדוגמה, סינון אלפי גנים בעלי שונות ביטוי נמוכה במרחב הפריטים), עם זאת, באופן מפתיע, מספר מועט מאוד של אלגוריתמים הוצעו לבחירת תכוניות באופן בלתי מפוקח. רוב השיטות הקיימות כיום הינן נאיביות באופיין, לדוגמה טווח ערכים, יחס בין ערכי מקסימום למינימום, בחירת תכוניות עם ערכים מעל סף תחתון מסוים, אנטרופיה או שונות. המשותף לשיטות אלו הוא שערכי כל תכונית (למשל שונות) מחושבים באופן בלתי תלוי באחרות.

חשיבות הנושא והמחסור בשיטות לא-מוטות, יציבות, ויעילות הובילו אותנו לפתח אלגוריתם לסינון בלתי מפוקח של תכוניות בשם UFF (Unsupervised Feature Filtering). UFF שונה מסכמות בחירה לא מפוקחות בשני מובנים: *(1)* הוא אינו מערב פונקציית מטרה כמדד לבחירה ו *(2)* הוא משקלל את ההשפעות ההדדיות של כל אחת מן התכוניות. UFF מנקד כל תכונית לפי תרומתה לאנטרופיית ה SVD של מערך הנתונים. תרומה זו נמדדת לפי עקרון ה 'השאר-אחד-בחוץ' (leave-one-out).

ישום השיטה במספר רב של בסיסי נתונים, מסוגים שונים (למשל ביטוי גני, נפיצות חומצות אמינו ברצפי חלבון) הראה כי: *(1)* בחירה של מספר קטן מאוד של תכוניות מסייעת לשיפור איכות האשכול (בהשוואה לקבוצות אחרות, בגודל דומה, שנבחרו ע"ס שיטות לא מפוקחות אחרות, או בהשוואה לאשכול הפריטים, כשאלו מיוצגים ע"י כל התכוניות); *(2)* גישת ה UFF נמצאה כעמידה גם במצבים של איבוד אינפורמציה חמור; *(3)* התכוניות שנבחרו ע"ס השיטה נמצאו כבעלות חשיבות

תקציר

רבים מכנים את התקופה הנוכחית כ״מהפכה הגנומית״. בשנים האחרונות גנומים של מאות אורגניזמים רוצפו, גנים של אותם אורגניזמים מופו ותוצריהם (מולקולות RNA, וחלבונים) נלמדו. מבניהם המרחביים של עשרות אלפי חלבונים נובא ונצפה. הפונקציות התאיות של כל אותן מולקולות מתבהרות בקצב מואץ.

הבסיס להאצה זו נעוץ במספר פריצות דרך טכנולוגיות, אשר מאפשרות ריצוף מהיר של מולקולות (לדוגמה Haplotype Map), מדידת ביטוי גנומי (לדוגמה מערכי DNA, או Comparative Genomic Hybridization), מדידת הקישוריות בין חלבונים לDNA (לדוגמה ChIP-on-chip), תכונות הפרוטאום (לדוגמה מערכי חלבונים, Mass Spectrometry) ועוד. המשותף לכל אותן טכנולוגיות הוא יכולתן למדוד בו זמנית עשרות ומאות אלפי נתונים. כלים אלו, שפיתוחם הואץ ע״י חברות מסחריות, הפכו את איסוף הנתונים לפשוט, אמין, מהיר וזול.

במקביל לשיפורים ביכולת איסוף הנתונים, התפתחויות טכנולוגיות בעיבוד, אחסון והעברת מידע, הקלו על אפסון הנתונים ושליפתם. כתוצאה מכך, קיימת כיום גישה חופשית למספר רב של מאגרי נתונים גנומיים באינטרנט (לדוגמה, NCBI, Stanford Genomics, Ensemble ו UniProt).

נגישותו של מידע זה, וכמויותיו הגדולות, פתחו צוהר לשאלות רבות ולכיווני מחקר חדשים, אשר נחשבו דמיוניים אך לפני שנים ספורות. רבים ממאמצי המחקר משויכים לדיסציפלינה מדעית חדשה, הנקראת ״ביולוגיה מערכתית״. ביולוגיה מערכתית קשורה קשר הדוק לתחום הביואינפורמטיקה, שהתפתח אף הוא בשנים האחרונות. בעזרת מחקרים בביולוגיה מערכתית, מנסים החוקרים להסיק על הקשרים ההדדיים בין מולקולות בתא (גנים, RNA, חלבונים ומטבוליטים), ללמוד כיצד קבוצות מסוימות של מולקולות משפיעות על תהליכים ביולוגים, וכיצד גורמים סביבתיים ומטבולים מעצבים את המערכת האקולוגית בתא.

השלב הראשון בדרך למתן מענה לשאלות השאפתניות, המוצגות לעיל, הינו חשיפת תבניות חבויות מתוך אותם ׳ענני נתונים׳. שיטות, השואפות לחלץ מידע מכמויות נתונים גדולות מאוד, מוגדרות כטכניקות של כריית מידע. כריית מידע לרוב, משלבת עקרונות מתמטיים, סטטיסטיים ולמידה חישובית. אלגוריתמים של כריית מידע המיושמים באנליזות בביואינפורמטיקה הינם *1)* שיטות כלליות שהותאמו לסוג הנתונים או *2)* אלגוריתמים שפותחו במיוחד כדי להתמודד עם שאלות מן התחום. דוגמאות לאלגוריתמים השייכים לקבוצה הראשונה, הינן רוב שיטות בחירת התכונות (feature selection, ראה להלן), אִשכול (clustering) ומיון (classification). דוגמאות לשיטות שפותחו בתחום הן אלגוריתמי אשכול למערכי גנים CLICK, CAST ו Gene-Shaving, ואלגוריתם BLAST להשוואת רצפים. אלגוריתמים השייכים לקבוצה הראשונה, הינם מאוד כוללנים, ויתכן שאינם מתאימים לטיפול באתגרים המיוחדים שהמידע הגנומי והחלבוני מציבים, ואילו אלגוריתמים מן הקבוצה השנייה, הינם תלויי תחום, ונראה שאינם מתאימים לפתרון בעיות מתחומי מחקר אחרים.

שלמי תודות

עבודה זו נעשתה בהדרכתם של פרופסור מיכל ליניאל ופרופסור דוד הורן

כריית מידע גנומי ופרוטאומי: אלגוריתמים, כלים ופרשנויות

חיבור לשם קבלת תואר דוקטור לפילוסופיה

מאת

רועי ורשבסקי

הוגש לסינט האוניברסיטה העברית בירושלים,

דצמבר 2007