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,
[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
*To whom correspondence should be addressed. royke@cs.huji.ac.il

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

[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

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.

arranging elements (data-points) into specific groups. Clustering, being an unsupervised learning problem, may be regarded as a special case of categorization with unknown 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

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

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 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

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

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 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]:

$$Jaccard = \frac{n_{11}}{n_{11} + n_{01} + n_{10}} \qquad \textbf{(4)}$$

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, 2 4] 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.

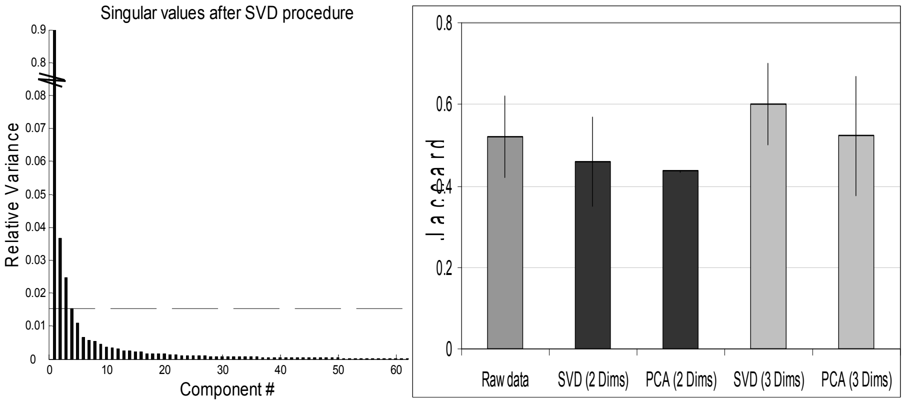

Fig. 1. A. (left) Singular values of the colon dataset (dashed line denotes the 'cut' decision). **B**. (right) Jaccard scores of the KM on raw data (left bar) and different preprocessing options.

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

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

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

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.

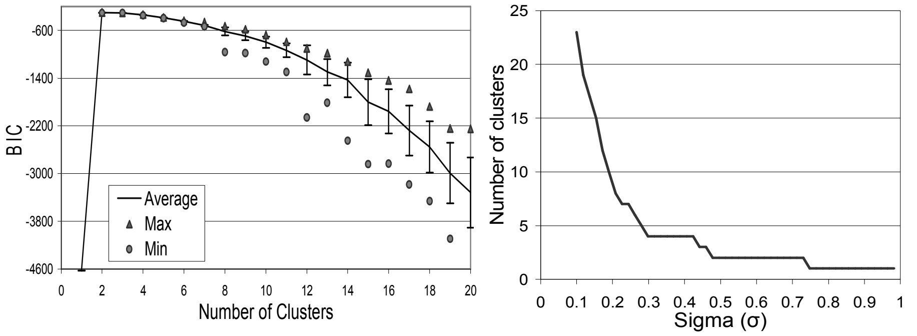

Fig. 2. A. (left) BIC Values when applying OKM (SVD reduced to 3 dimensions) on the colon dataset. **B.** (right) The number of clusters obtained in the colon dataset as a function of the σ input parameter of the QC algorithm.

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.

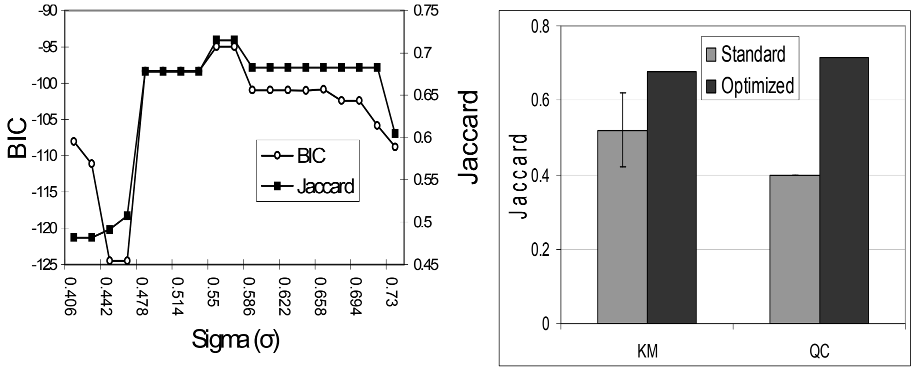

Fig. 3. A. (left) Comparison of the internal (BIC) and external (Jaccard) criteria for the colon dataset (OQC). **B.** (right) Comparison of the standard and optimized versions of the KM and QC algorithms

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 $\sigma=0.548$, 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 $0.47<\sigma<0.56$.

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

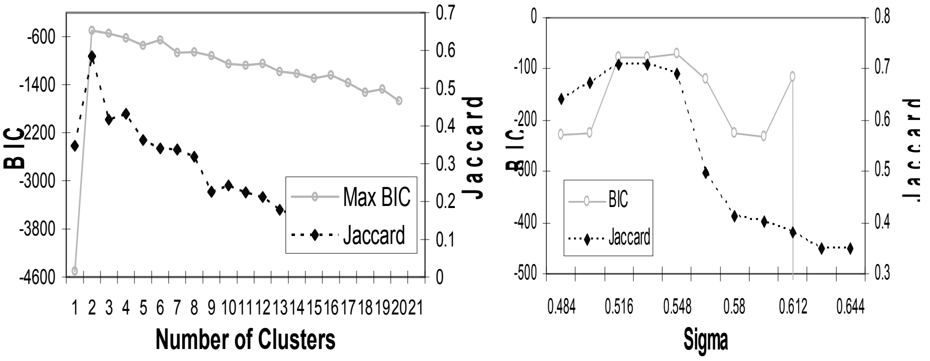

Fig. 4. A. (left) BIC and Jaccard scores of the Golub dataset (OKM), **B.** (right)Comparison of internal (BIC) and external (Jaccard) criteria of the leukemia dataset (OQC)

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.

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. Conclusions

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.

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.