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 classification 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 dimensional reduction and the second is an automated tuning of the algorithm's parameter(s). The dimensional 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).

Introduction[1]

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

[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. awarded a fellowship from the SCCB, the Sudarsky Center for Computational Biology. This work is partially supported by EU Framework VI, DIAMONDS consortium.

algorithms we will expand on is Quantum Clustering (QC), the effectiveness of which was demonstrated on gene-expression data [7], [8].

For large scale gene-expression data, 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 work for all kinds of data. Different preprocessing methods suggest averaging and variance standardization, excluding genes with low variance between conditions [2], PCA, Fourier transforms [12], and more.

One fundamental preprocessing direction is dimensional reduction. [13] claim that the dimension should be correlated with the expected number of clusters. 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. Clearly, in cases that the number of clusters is a priori unknown, this procedure may be useless.

Several efforts to develop efficient and accurate filtering schemes and compression tools were 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 that have a variance below a certain threshold between different experimental conditions are discarded. Obviously, such filtration 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, it seems impossible to assess the correct number of clusters (e.g., as required by the K-Means algorithm). 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 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.

[2] In the paper N is referred as the number of elements in the data, and K as the number of clusters.

In addition, algorithms such as K-Means, Fuzzy C-Means and others, being nondeterministic, are inconsistent as they are susceptible to starting points and other stochastic factors. Some methods such as averaging clustering results, following a majority rule, or applying different heuristics [16] were introduced.

Since different results may be obtained by the numerous clustering algorithms that are used, 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 that embed an internal criterion that can be incorporated in any clustering algorithm.

Methods

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

Preprocessing

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

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 dimensionality 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 sample and gene 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 of 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 1 and supplementary), one may come up with several principles for truncation.

$$V_i = \frac{s_i^2}{\sum_j s_j^2} \quad \textbf{(1)}$$

[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 may use Shannon's entropy [19] as a guide for choosing among the possibilities.

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

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, therefore renormalization is performed by projecting the vectors onto the unit hyper-sphere in r-space. This approach is consistent with the standard application of LSA (Latent Semantic Analysis), where similarity between vectors in the truncated space is defined in terms of the cosine of the angle between them [20]. Clustering is then performed on the hyper-sphere, where all data points are represented by vectors with equal norms.

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 that 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:

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

where $l_M(x,\Theta)$ is the mixture log likelihood 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, [10] and other have applied EM algorithm to find a partition that maximizes the BIC criterion. In our method we do not optimize the BIC score but instead we use the BIC to measure the clustering results.

Implementation

We implement the framework for two fundamentally different clustering algorithms that serve as representatives for other clustering algorithms (such as listed in above). The two representative algorithms differ in some fundamental aspects thus testing the generality of our framework.

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

Optimized K-Means (OKM)
K-Means is a very popular, fast and intuitive algorithm. This naïve algorithm has some known disadvantages that impair its performance. 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 susceptible to inconsistency.

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 implementation of the framework optimizes the QC algorithm. The algorithm [7] uses the Schrodinger 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. 4). Contrary to K-Means this algorithm is deterministic, has less constraints than K-means (since it integrates the presence of noise in its model), and does not assume spherical properties of the clusters.

The OQC implementation applies the QC algorithm once for a range 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 the optimal solution.

Results

Here we describe our results on three gene-expression datasets that are considered to be benchmarks in the field. In the first [22] and the second [23] examples samples were clustered (2 and 4 clusters, respectively) while in the third dataset [24] clustering was performed on the genes. All three cases have assignments that were manually obtained by experts. 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]:

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

1. The colon dataset of Alon et. al. (1999)
In the dataset of [22], 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 [22], [2] and [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) and PCA (Principal Components Analysis) with

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

standardized variables, based on correlations. Fig 1 displays the singular values of the [2000x62] matrix.

Following 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: 0.284, see Formula 2). This yields compression rates of $1x10^{-3}$ and $1.5x10^{-3}$, respectively (see supplement for a complete description),

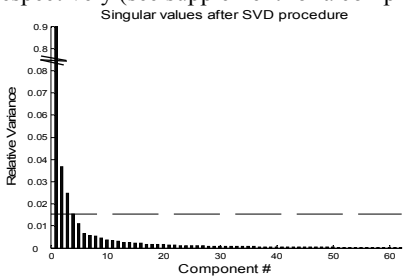


Fig. 1. (left) Singular values of the colon dataset

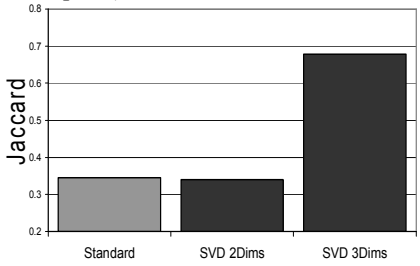
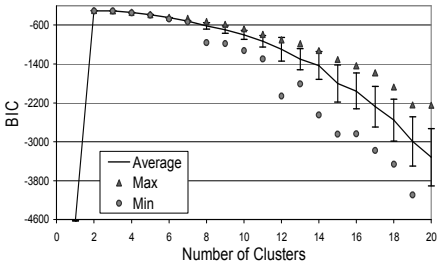


Fig. 2. (right) Jaccard scores of the KM (left, gray bar) and OKM algorithms (black bars) following different preprocessing methods.

As displayed in Fig. 2, the preprocessing procedure significantly influences the clustering quality. We therefore conclude that this step deserves substantial attention. Moreover, when selecting the correct compression method (SVD, in this case) and extent, the clustering results are significantly improved, as reflected by the increase in the Jaccard score from 0.34 to 0.678. The Jaccard score is only marginally improved by a PCA-based process (not shown).



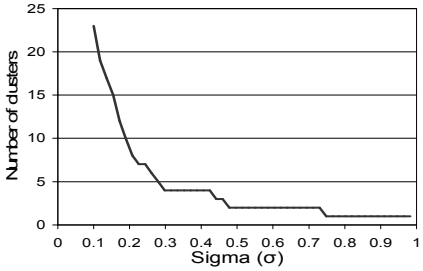


Fig. 3. (right) The number of clusters obtained in the colon dataset as a function of the σ input parameter of the QC algorithm

Fig. 4. (left) BIC Values when applying OKM (SVD reduced to 3 dimensions) on the colon dataset

The optimal results for the colon dataset were obtained for SVD reduction to 3 dimensions. At this stage, the data are compressed to 62 vectors on a 3 dimensional unit sphere. Fig. 3 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.

As displayed in Fig. 3, the BIC is maximized where two clusters describe the dataset. The observation also shows that the farther the number of clusters is from the correct solution, the larger is the BICs' dispersion. One can note that although the BIC's averages are very similar, the BIC of 2 clusters is maximal. 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.

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. 4 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.

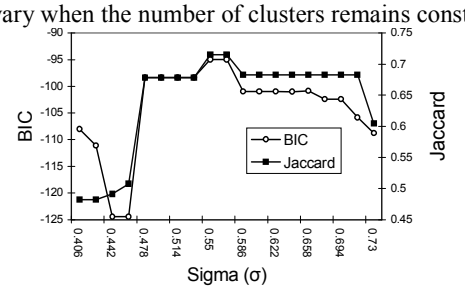


Fig. 5. (left) Comparison of the internal (BIC) and external (Jaccard) criteria of the colon dataset (OQC)

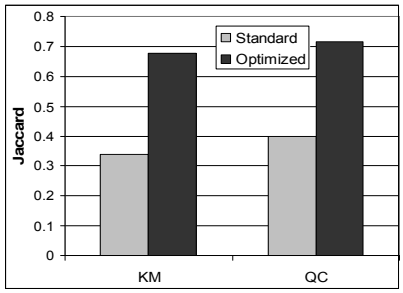


Fig. 6. (right) Comparison of the standard and optimized version of the KM and QC algorithms

Comparing the BIC and Jaccard scores in the neighborhood of their maximal values shows a nearly perfect match (Fig. 5). The maximum BIC was obtained for σ = 0.55, which dictates 2 clusters. The corresponding Jaccard score for this σ is 0.715.

Since the OKM and OQC both share the same preprocessing step, their clustering results can be compared. The maximal BIC value achieved by the OQC is higher than the one achieved by the OKM (-95 and -300, respectively). Similarly, the corresponding Jaccard of the OQC is higher than the one of OKM (0.715 and 0.678, respectively). In both cases, the suggested framework demonstrates its effectiveness when comparing these results with what the same algorithms obtain on the original datasets (0.678 vs. 0.34 for KM and 0.715 vs. 0.4 for QC, respectively, see Fig. 6).

2. The Leukemia dataset of Golub et al., 1999

The dataset of [23]. has served as a benchmark for several clustering methods [2], [4]. 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]).

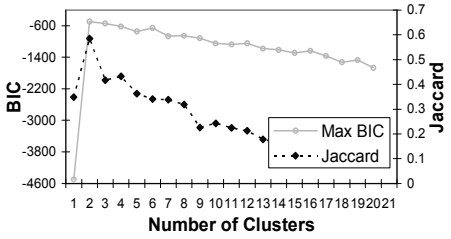


Fig. 7. (left) BIC and Jaccard scores of the Golub dataset (OKM)

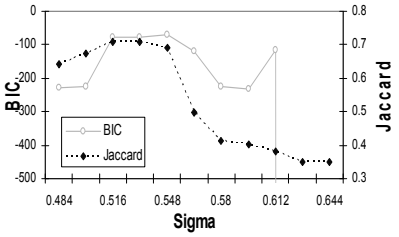


Fig. 8. (right)Comparison of internal (BIC) and external (Jaccard) criteria of the leukemia dataset (OQC)

Exercising the same protocol as in the dataset described above yields similar results. BIC is maximized at two clusters, as is the Jaccard score (Fig. 7). Hence we conclude that OKM can identify only the two major groups in the data and cannot detect a partition to four groups. This finding is consistent with the CAST and CLICK algorithms that also failed to identify the subtypes [2]

Since QC can not be applied to the raw dataset, preprocessing is of the essence. As a result, exercising the OQC on the dataset proves to be very effective. As displayed in Fig. 8, 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. 8).

3. The Yeast dataset of Spellman et al. (1998)

The dataset of [24] 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 the relative variances is more gradual and the entropy is significantly higher (0.705, see Formula 2 an 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 suggests that there are 5 groups of cell cycle related genes. When applying the OKM protocol on the compressed dataset a maximized BIC is observed at 6 clusters. Comparing to the standard application of K-Means, the OKM shows no improvement, as both applications yield Jaccard scores of 0.4.

Application of the OQC procedure on 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 jointed into one cluster.

In the current dataset, the fit between the BIC and the Jaccard scores is not perfect (see supplementary). Nevertheless, the OQC shows a significant improvement in comparison with the standard application of the QC algorithm (Jaccard scores of 0.5, and 0.4, respectively).

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 dimensionality reduction, several iterations of a clustering algorithm are applied, each with a different parameter. They are then compared with each other by BIC criterion. The parameter that yields the best BIC score is chosen and its associated parameter is declared as the optimal one. This generic framework is also computational efficient, as 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).

Unarguably, preprocessing of experimental data is an essential step. The raw data, often comes in a large-scale, un-normalized and noisy representation. These distractions have to be treated. Nevertheless, due to the diversity of the experiments one is preempted from providing an absolute recipe for such a preprocessing. In our study, we emphasize the important role of the preprocessing and the compression in particular, and present some examples of the variations that different preprocessing methods can yield. Among the preprocessing methods we suggest application of the 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 the framework 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 algorithm.

We find application of BIC useful in finding the best solution amongst many local maxima, in deterministic and nondeterministic algorithms. Some heuristics are proposed in order to overcome the inconsistency problem of nondeterministic algorithms. Nevertheless, contrary to averaging, majority voting and others suggestions [16] our global maximum search can hopefully identify the best solution among many local maxima. In cases where many applications of the same algorithm suggest suboptimal solutions and only a few suggest the optimal one, BIC maximization search overcomes the needs for a majority vote.

The framework is especially well adopted by algorithms that assume spherical distribution (e.g., K-Means) of the clusters, though it can be applied to algorithms that do not assume such a distribution. Surprisingly, its best performance is obtained when applying an algorithm that has fewer constraints in its model e.g., QC (this work) and SOM (not shown), and does not assume any structured distribution.

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

Second, the BIC score assumes a specific hyper-elliptical organization of the clusters. When, as in the yeast dataset, clusters have different distributions, BIC has less descriptive strength. In those cases the level of fitting between BIC and the underlying properties of the dataset is reduced. Third, the BIC, computed by the EM method, usually can not converge when the number of dimensions rises above a very limited number (practically, less than a dozen). 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 can not compare models of different datasets. Therefore, the measurement can not be applied to different preprocessing methods or dimensions, since they reflect different distributions of the dataset. We suggest taking these limitations under consideration when utilizing BIC as a comparison tool.

Recently, a BIC implementation for gene-expression data was demonstrated [10]. Ttwo variations of the maximization of the criterion on a mixture of two distributions were compared (containing mostly simulated data). The presented method partitions the samples into two clusters when two gene-expressions are examined (i.e., two dimensions). Alternatively, we suggest computing the BIC score on a clustering algorithm's result. Specifically, scanning the entire parameter-space for a given algorithm and mapping the maxima of BIC. The motivation for this variation is two-fold: *(a)* this suggests a more generalized utilization of the BIC criterion, by allowing it measure any clustering algorithm. *(b)* Maximization of the criterion, using the EM algorithm is computationally expensive (see Supplementary). Hence, computing the BIC on the preprocessed, compressed set is a prominent advantage where large-scale datasets are analyzed.

Herein, we present a general 'clustering improver' scheme, and demonstrate how it can leverage simplistic and inefficient algorithms with poor performances to favorably compete with more sophisticated, state-of-the-art clustering algorithms. The optimized algorithms described here outperform the published results of CTWC, CLICK and CAST (for a detailed comparison see supplementary). We assume that applying this framework to the latter algorithms could improve their performance even further. Furthermore, we also find that when BIC does not give the best solution, it gives one that is close to the best. It therefore can assist in narrowing down the search (e.g., yield a range of the number of clusters or σ values).

Finally, different clustering algorithms are currently included in analysis suites that are applied by experimentalists for gene expression data. 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 gene-expression and other datasets are susceptible to objectivity bias. 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 edn: 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. 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.
23. 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.
24. 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.