Clustering gene expression data with the PKNNG metric.

Ariel E. Bayá, Pablo M. Granitto

CIFASIS, CONICET-UNR-UPC/Marselle
Bv 27 de Febrero 210 Bis
Rosario, 2000 (República Argentina)
{baya,granitto}@cifasis-conicet.gov.ar

Abstract

In this work we use the recently introduced PKNNG metric, associated with a simple Hierarchical Clustering (HC) method, to find accurate an stable solution for the clustering of gene expression datasets. On real world problems it is important to evaluate the quality of the clustering process. According to this, we use a suitable framework to analyze the stability of the clustering solution obtained by HC+PKNNG. Using an artificial problem and two gene expression datasets, we show that the PKNNG metric gives better solutions than the Euclidean method, and that those solutions are stable. Our results show the potential of the association of the PKNNG metric based clustering with the stability analysis for the class discovery process in high–throughput data.

1 Introduction

Clustering is a fundamental topic in machine learning and pattern recognition. Its final aim is to find any arbitrary structure hidden in a set of data, which is critical in biological applications like microarrays data analysis [3]. In those cases, when researchers evaluate thousends of genes at once, it is important to provide them with analysis tools that can help to understand the data [1, 3].

In a previous work we introduced the Penalized K–Nearest–Neighbor–Graph based metric (PKNNG)[16], a new method capable of finding clusters located on non-linear manifolds (non-linear low dimensional surfaces embedded in high dimensional spaces). PKNNG follows the idea behind ISOMAP [8], Locally Linear Embedding [9] or Laplacian Eigenmaps [11], looking for local neighborhood relations that can be used to produce low dimensional projections of the data at hand. The new metric naturally extends the application of most previously introduced clustering methods [4] to these cases. The PKNNG algorithm has two stages. Following ISOMAP, it first searches for locally uniform manifolds (which could be disjoint) and then a connection algorithm is used to group the disjoint manifolds found in the first stage. Using three artificial problems we showed that the method can easily find cluster with arbitrary shapes in high dimensional datasets.

The main drawback of clustering methods is that they always find a data grouping, even when there is none. We need methods that can find **natural groupings**, the structures that can be truly inferred from the data and not obtained as an artifact of the clustering algorithm. Unfortunately, there is no general consensus yet of the definition of natural groupings, but several relevant works [12, 7, 13] relate the concept with clustering solutions that are highly stable under small perturbations. Ben-Hur & Guyon [12] introduced a method for assessing stability, based on clustering perturbed versions of the dataset under analysis and evaluating the stability of the solutions. Using artificial and real world examples, the authors showed that their algorithm is a valid method for detecting stable structures, also detecting the lack of structure in the data. Monti et. al. [13] used a similar concept, also showing good results, but their method was developed as a visual tool.

Inherently hierarchical algorithms (HC) [5] are more stable than partitional algorithms. Divisive HC methods have a "bottom-up" approach to construct a dendrogram, where each level of the dendrogram represents a particular clustering of the data. Thus, consecutive levels of the dendrogram are related. Partitional algorithms [6], on the other side, determine a fixed number of clusters, all at once starting from k random clusters, searching iteratively for a locally optimal solution of the clustering problem. As a result, solutions with consecutive k are not related as in dendrogram.

In this work we evaluate the possibility of using the new PKNNG metric to find natural groupings in gene expression datasets. We couple the new metric with a hierarchical clustering method, in order to find more stable solutions. We evaluate the stability of our clustering solutions using the procedure introduced by Ben-Hur & Guyon. We show the potential of this setup with an artificial dataset, and then we apply it to find natural groupings in two gene expression datasets.

The rest of this paper is organized as follows. In Section 2 we review the Isomap-based method to construct a fully connected non-linear manyfold, the PKNNG metric, and we discuss in detail the stability analysis developed by Ben-Hur & Guyon. In Section 3 we apply this setup to cluster the three datasets and evaluate their stability, and also we compare our results to those previously obtained with other methods. Finally we draw some conclusions and discuss future lines of research.

2 Methods

2.1 The PKNNG Metric

In previous works [15, 16] we introduced an ISOMAP based metric that is useful to cope with clusters of arbitrary shape. The method follows the idea behind Isomap [8], which states that in a curved manifold the geodesic distance between neighbouring points can be correctly approximated by the Euclidean input space distance, but for faraway points geodesic distances are better approximated by adding a series of short hops between neighbouring points.

In Table 1 we show the PKNNG algorithm. PKNNG takes as inputs a dataset, a given connection method and the value of k, the number of neighbours to be used, and outputs a distance matrix, which is constructed measuring distances in a specifically created graph.

Input: a Dataset {Data}, {k} the number of neighbours and {method} a connection method

Output: {D} the distance matrix.

Procedure:

1. Obtain the k–nearest–neighbours–graph using K neighbours: **KnnGraph** $= Knng(Data, k)$
2. Remove outlayers and symmetrize: **KnnGraph** $= Clean(KnnGraph)$
3. Connect the graph with the selected method: **GraphPKNNG** $= connect(KnnGraph, method)$.
4. Calculate all pairs distances using the graph: $\mathbf{D} = Distances(GraphPKNNG)$

Table 1: The PKNNG algorithm

As a first step, the method searches for locally dense structures. The goal of this stage is to obtain several disjoint structures, where each structure gather highly similar points. To this end, PKNNG constructs the k–nearest–neighbours–graph of the data, i.e. the graph with one vertex per observed example, and arcs between each vertex and its k near neighbours with weights equal to the Euclidean distance between them[1]. Then, using an appropriate strategy [16], we add edges with a penalized metric, in order to connect all structures, giving as result a single connected graph. Using this graph we can now compute geodesic distances between faraway points using computational efficient algorithms like Floyd or Dijkstra [10].

As we mentioned before, after step 2 in Table 1 we can have several disjoints subgraphs. The number of structures and their connection degree are directly related to the number of neighbours k used to construct the knn–graph. In all our previous simulations [16] we verified that this method captures the true topology of the data for a wide range of values of k. We also verified that the key factor of the method is the use a penalized metric for the edges added in the step 3 of Table 1:

$$w = d \ \ e^{d/\mu}, \tag{1}$$

where w is the graph weight corresponding to the added edge between structures, d is the Euclidean distance between the vertices being connected by that edge and μ is the mean Euclidean distance between nearest neighbours in the graph. For the purpose of this work we use the *AllSubGraphs* connection method [16], which connects each structure to all the remaining structures through their nearest pair of points, of course using the penalized metric.

2.2 Stability

In this section we present the stability analysis introduced by Ben-Hur & Guyon [12]. The method is based on a simple idea: If a problem has a natural grouping, we should be able to arrive to that solution starting from perturbed versions of the dataset. Or, equivalently, if we found the same solution starting from slightly diverse datasets, that solution should not be an artifact introduced by the clustering method. They propose to create perturbed datasets by sub-sampling the original data, cluster each one of them, and measure how similar the diverse

[1] After this process we eliminate outliers from the graphs. We consider that an arc is an outlier if it is not reciprocal (i.e. one of the vertex is not a k-nn of the other) and the length of the arc is an outlier of its distribution (i.e. if it is bigger than the 3rd quartile plus 1.5 times the inter-quartile distance of its distribution).

clustering solutions are. The authors suggest to evaluate solutions with a growing number of clusters and to select the stable solution with the biggest number of clusters.

In Table 2, we present a high level pseudo-code of the stability algorithm. The inputs of the algorithm are *Data*, which is the Dataset to be clustered, K_{max}, the maximum number cluster to consider and *Rep*, the number of resamplings of the dataset to use for each k. The procedures outputs $S(i,k)$, which is a list that for every k contains *Rep* similarities scores. The method itself starts at *line 1* by defining f which is the size of the sub-samples of *Data* that will be using. *Line 2* sweeps all values of k from 2 to K_{max}, then *line 3* repeats *Rep* times the operations made for each k of *line 2*. This operations consist of taking two sub-samples of *data*: sub_1 and sub_2, clustering them and then obtaining labels L_1 and L_2 respectively. From sub_1 and sub_2 we can calculate the intersection points and then we can measure their similarity using $s(a,b)$.

Assume that is given a dataset $\mathbf{X} = \{x_1, x_2, x_3, ..., x_n\}$, where $x_i \in \mathbb{R}^d$. The labeling $\mathcal{L}$ defines k partitions in $\mathbf{X}$ (for example, $\mathcal{L}$ can be a clustering method that produces k non-overlapping partitions $S_1, S_2, ..., S_k$ of the dataset). Then we define a matrix C $(n \times n)$ where:

$$C_{i,j} = \begin{cases} 1 \text{ if } x_i \text{ and } x_j \text{ belongs to the same cluster} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

Two labellings $\mathcal{L}_1$ and $\mathcal{L}_2$ have a corresponding pair of matrices $C^{(1)}$ and $C^{(2)}$. The dot product of this pair of labellings would be:

$$\langle \mathcal{L}_1, \mathcal{L}_2 \rangle = \langle C^{(1)}, C^{(2)} \rangle = \sum_{i,j} C^{(1)}_{i,j} \cdot C^{(2)}_{i,j} \tag{3}$$

This dot product represents the common edges in a graph represented by $C^{(1)}$ and $C^{(2)}$, which also tells as which pairs of points are clustered together. As a dot product $\langle \mathcal{L}_1, \mathcal{L}_2 \rangle$ satisfies the following inequality: $\langle \mathcal{L}_1, \mathcal{L}_2 \rangle \leqslant \sqrt{\langle \mathcal{L}_1, \mathcal{L}_1 \rangle \cdot \langle \mathcal{L}_2, \mathcal{L}_2 \rangle}$ and so we can derive a normalized form:

$$cor(\mathcal{L}_1, \mathcal{L}_2) = \frac{\langle \mathcal{L}_1, \mathcal{L}_2 \rangle}{\sqrt{\langle \mathcal{L}_1, \mathcal{L}_1 \rangle \cdot \langle \mathcal{L}_2, \mathcal{L}_2 \rangle}} \tag{4}$$

where equation 4 is a correlation similarity measure.

The only problem remaining is that the same cluster can be assigned a different arbitrary number by two different labellings. Altough we follow the framework presented by Ben-Hur & Guyon in [12] to this point, they used an aproximated method to solve this problem. Instead, we choose to use the exact value, which only requires more computation [12].

Also following Ben-Hur & Guyon [12], we present the results (the scores corresponding to the set of *Rep* similarities for each possible value of k) as plots of cumulative distribution functions (CDF). Stable solutions are functions located near the right–bottom corner of CDF plots (with high similarities in almost all runs), and unstable solutions lies near the top–left of the plots. The idea is that it should be a noticeable gap between the set of CDF curves corresponding to stable solution and the set corresponding to incorrect solutions.

Input: a Dataset $\{Data\}$, $\{K_{max}\}$ the maximum number of clusters and $\{Rep\}$ the number of repetitions of the sampling procedure.

Output: $\{S(i,k)\}$ a list of $\{Rep\}$ similarities for each k, where $i = 1, 2, ..., Rep$ and $k = 1, 2, ..., K_{max}$

Procedure: $cluster(X, k)$ is a clustering algorithm that takes as input parameters a Dataset X and k a number of clusters. $s(Set_1, Set_2(Intersect))$ a similarity measure between two sets

1. $f = 0.8$
2. **for** k in 1 to K_{max}
3. **for** i in 1 to Rep
4. sub_1 = sample fraction f of **Data**
5. sub_2 = sample fraction f of **Data**
6. $L_1 = cluster(sub_1, k)$. Cluster solution on subsample 1 using k clusters.
7. $L_2 = cluster(sub_2, k)$
8. $Intersect = sub_1 \cap sub_2$
9. $S(i,k) = s(L_1(Intersect), L_2(Intersect))$. Computation of similarity on the intersection of sub_1 and sub_2.
10. **end for**
11. **end for**

Table 2: Stability algorithm.

3 Results and Discussion

In this section we report the results of applying our method to three different datasets, one artificial and two real. In all three cases we know the true classes of the data and we suppose that the natural grouping is represented by these classes. We always compare the structure found by the clustering algorithm with the original classes using confusion matrices. Also, we analyze the stability of the solutions using the procedure described in Table 2. In all experiments we set $f = 0.8$ and $Rep = 100$. As clustering algorithm we use HC with average linkage [5]. HC has an unwanted effect, it sometimes produces singleton clusters. To solve this problem we stablished a threshold of 3 points as the minimum numbers of elements that is considered to form a cluster.

3.1 Three–Rings

This is an artificial two dimensional dataset composed by 1200 points. As can be seen on Figure 1, this dataset has five true clases, each one represented by a different colour.

We clustered the dataset using the PKNNG metric and the clasical Euclidean metric. In figure 2 we show the stability analysis for PKNNG (left panel) and Euclidean metric (right panel). For PKNNG there are stable structures for $k = \{2, 3, 5\}$. For $k = 2$ the algorithm separates the black cluster at the center from the other 4 clusters, for $k = 3$ the clusters correspond to

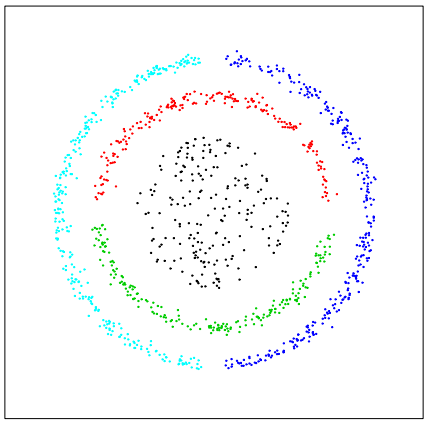
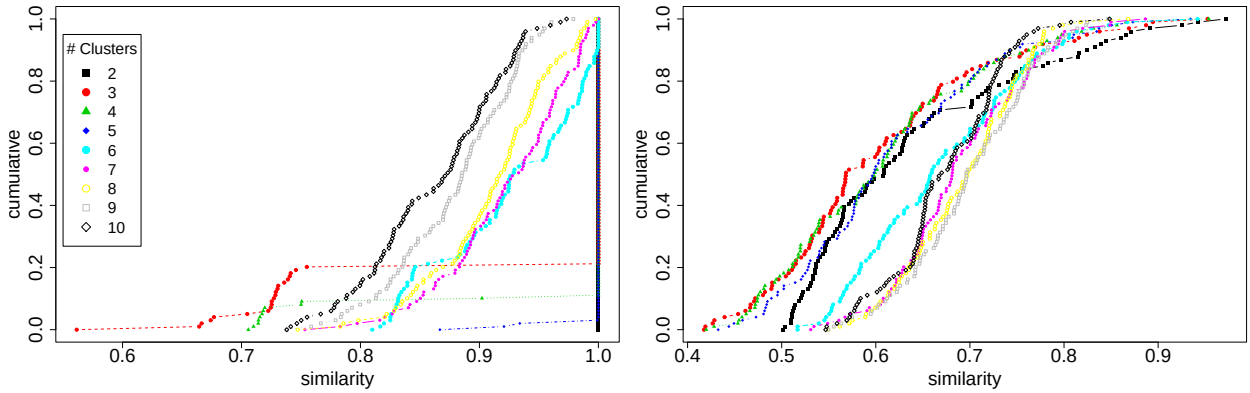


Figure 1: The Three–Rings dataset.



(a) PKNNG (b) Euclidean

Figure 2: Stability analysis for the Three–Rings Dataset. Left panel: PKNNG metric. Right panel: Euclidean metric.

the three rings and, finally, for $k = 5$ HC with PKNNG metric finds the right five clusters. As it is stated on Ben-Hur & Guyon [12] k is chosen as the bigest value that shows good stability. In this example $k = 5$ is the right solution. Bigger values of k ($k \geq 5$) are considerably less stable. For the Euclidean metric (right panel) we can see that CDF curves for all values of k are tangled. There is no stable solution in that case.

In Table 3 we show the corresponding confusion matrices for five clusters, which is the stable solution for HC+PKNNG-metric and also the true number of clusters. It is clear from the tables that the stable solution found by HC+PKNNG-metric is the right solution, and that HC cannot find an appropriate clustering using the Euclidean metric.

3.2 Yeast

The Yeast DNA dataset was introduced by Eisen et. al. [1], where they noted that this dataset clustered well. Subsequently, Brown et. al. [2] used MYGD functional annotations to select the most learnable examples by SVM according to 5 functional classes. As a result they obtained a five class dataset with 208 genes and 79 features (each feature correspond to an experiment, and

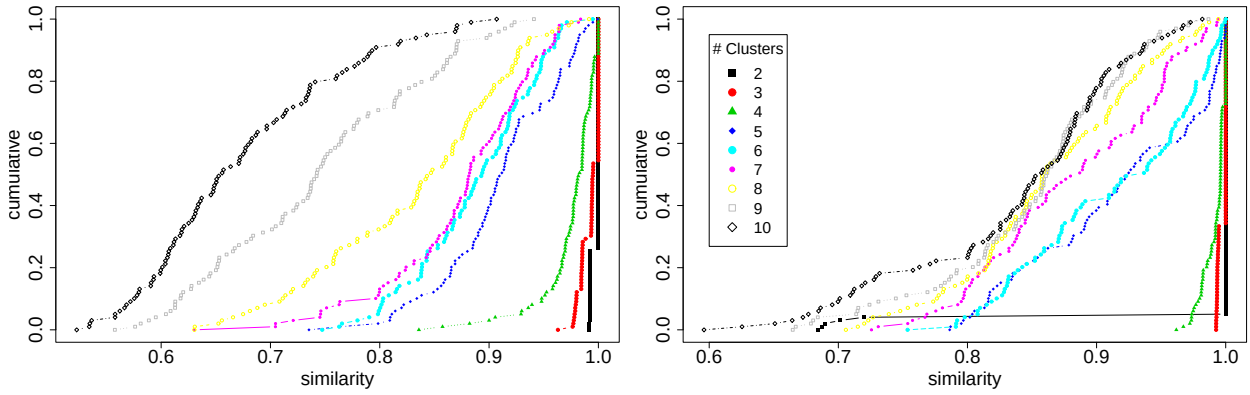
(a) Euclidean

	1	**2**	**3**	**4**	**5**
5	128	24	0	23	125
3	0	126	0	0	74
2	65	0	28	107	0
1	0	0	200	0	0
4	0	141	61	98	0

(b) PKNNG

	1	**2**	**3**	**4**	**5**
5	300	0	0	0	0
3	0	200	0	0	0
2	0	0	200	0	0
1	0	0	0	200	0
4	0	0	0	0	300

Table 3: Confusion matrices for the Three–Rings dataset. Rows correspond to the true classes, columns to the resulting clusters.



(a) PKNNG (b) PCA - 3 components

Figure 3: Stability analysis for the Yeast Dataset. Left panel: PKNNG metric. Right panel: Euclidean metric using the first 3 components of the PCA projection.

the goal is to cluster the genes). The five classes correspond to Tricarboxcylic Acid Cycle (TCA, 14 genes, class 0), Respiration chain complexes (27 genes, class 1), Cytoplasmaticribosomal proteins (121 genes, class 2), proteasomes (35 genes, class 3), and histones (11 genes, class 4).

In this case we compared HC+PKNNG-metric in the original 79–dimensional space with using HC with the Euclidean metric on the first three components of the PCA projection of the dataset. This last setting was found to be optimal in previos works on the yeast dataset [12]. Figure 3 shows the stability of both approaches. Analizing Panel a (PKNNG), we found a gap between the CDF for $k = 4$ and $k = 5$. According to this, there are stable clustering solutions for $k = \{2, 3, 4\}$ and we should choose $k = 4$ as the solution with PKNNG. Analizing panel b (Euclidean on PCA projection), we found the same kind of gap between CDFs at $k = 4$ and $k = 5$, so for this setting the problem solution is also $k = 4$. Table 4 presents the confusion matrices for both settings using four clusters. Both approaches show comparable performaces, though there are small differences. PCA Confusion matrix shows that this method missclasify two more patterns, one of class 3 and one of class 4, while PKNNG solutions presents two outlayers (in columns 5 and 6) that can not be considered as clusters, as we stated before.

3.3 Leukemia

This dataset, introduced by Golub et. al. [3], is a set of bone marrow samples prepared at the time of diagnosis: 11 samples of Acute Myeloid Leukemia (AML class), 8 of Acute

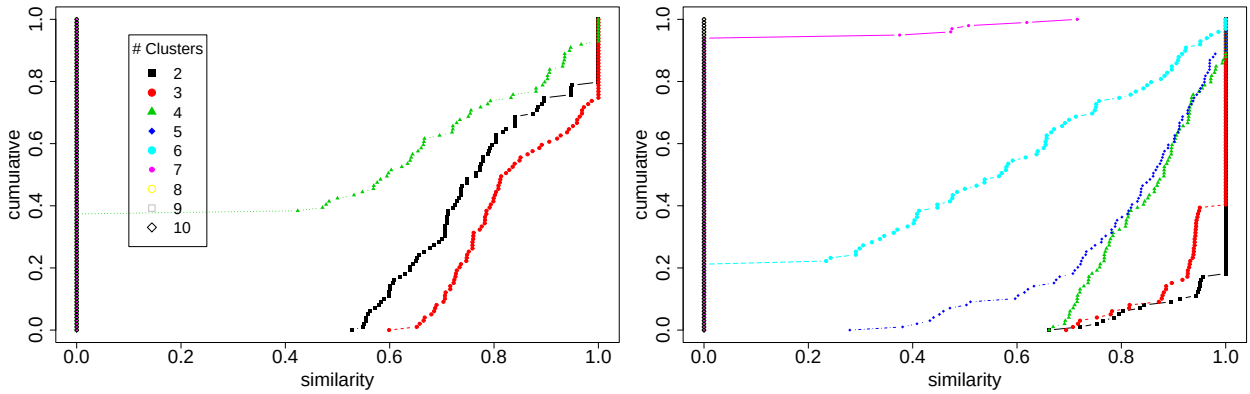
(a) PCA - 3 components

	1	**2**	**3**	**4**
3	8	0	3	0
2	0	121	0	0
4	0	0	32	3
1	0	0	0	27
0	0	0	2	12

(b) PKNNG

	1	**2**	**3**	**4**	**5**	**6**
3	9	0	2	0	0	0
2	0	121	0	0	0	0
4	0	0	33	2	0	0
1	0	0	0	27	0	0
0	0	0	0	12	1	1

Table 4: Confusion matrices for the Yeast dataset.



(a) PKNNG (b) Euclidean

Figure 4: Stability analysis for the AML-ALL Dataset. Left panel: PKNNG metric. Right panel: Euclidean metric.

Lymphoblastic Leukimia T-linage (T–ALL class) and 19 of the B-linage (B–ALL class). RNA prepared from bone marrow cells was hybridized to a Human Genome HU6800 Affymetryx microarray. From the 6817 genes present in the microarray we selected 1000 using the method described by Monti et. al. [13]. We centered the data (substracting the mean expression of each gene). The resulting dataset comprises 1000 genes measured on 38 patients, and the goal is to use the genomic expresion information to cluster the patients by their desease.

In figure 4 we present the stability analysis for this problem. In the left panel we show the results of HC+PKNNG-metric and in the right panel of HC with the Euclidean metric. PKNNG shows stable clustering solutions for $k = \{2, 3\}$, being $k = 3$ the actual solution. For Euclidean metric (the original method used by Golub et. al. [3]) we observe stable structures for $k = \{2, 3, 4, 5\}$ and the solution for this case is $k = 5$. This last result agrees to the one presented by Monti et. al. [13], although we applied a different normalization procedure.

(a) Euclidean

	1	**2**	**3**	**4**	**5**	**6**	**7**
0	17	1	1	0	0	0	0
1	0	0	0	8	0	0	0
2	1	0	0	0	6	2	2

(b) PKNNG

	1	**2**	**3**
0	18	1	0
1	0	8	0
2	1	0	10

Table 5: Confusion matrices for the AML-ALL dataset.

In Table 5 we compare the confusion matrices for both metrics. As we explained before, gropus with two or less samples are not considered as clusters, as for example columns 3 and 4 from the left Table. The results for PKNNG (right Table) represent a very accurate solution, which is very similar to the one obtained by Golub et. al. [3]. Both solutions (our and Golub's) incorrectly associates a B-ALL to a T-ALL cluster and an AML to a B-ALL cluster.

4 Conclusions

In this paper we applied the new PKNNG metric, coupled with a hierarchical clustering method, to find accurate and stable clustering solutions for two genomic expression dataset. After reviewing our metric and describing a simple method to evaluate the stability of a clustering solution (developed by Ben-Hur and Guyon), we used an artificial dataset to show the potential of these methods to find stable clustering solutions for problems where clasical Euclidean–metric–based solutions fail.

The results on the two dataset under analysis are encouraging. In the case of the yeast dataset, the PKNNG method found the same stable solution as the Euclidean metric evaluated on a PCA projection, and overall returns a slightly better clustering solution. The PKNNG method worked directly over the original space, avoiding the possible information loss associated with the linear PCA projection. For the AML-ALL dataset we obtained the right number of clusters as stable solution, where the original method (Euclidean metric) found more clusters. Evaluating the accuracy of both methods, again PKNNG produced a better clustering solution in this dataset.

Overall, these results show the potential of the association of the PKNNG metric based clustering with the stability analysis for the class discovery process in high–throughput data. As future work we plan to evaluate other datasets, and to use the full method (PKNNG metric plus stability analysis) in the search for reduced sets of genes that behaves in a coherent way (sometimes called metagenes).

Acknowledgements

We acknowledge partial support for this project from ANPCyT grants PICT 643 and 2226 (2006).

References

[1] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein, "Cluster analysis and display of genome-wide expression patterns", PNAS, pp:14863-14868, 1998.

[2] M. P. S. Brown and W. N. Grundy and D. Lin and N. Cristianini and C. Sugnet and T. S. Furey and M.Ares, Jr. and D. Haussler, "Knowledge-based Analysis of Microarray Gene Expression Data Using Support Vector Machines",PNAS,pp: 262-267,2000.

[3] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M. 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", Science, 286:531-537, 1999.

[4] R. Xu, and D. Wunsch II, "Survey of Clustering Algorithms", *IEEE Trans. on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[5] B. King, "Step-wise clustering procedures", *J. Am. Stat. Assoc.*, vol. 69, pp. 86–101, 1967.

[6] J. McQueen, "Some methods for classification and analysis of multivariate observations", *Proc. Fifth Berkeley Symp. on Math. Statistics and Probability*, pp. 281–297, 1967.

[7] R. Tibshirani, G. Walther and T. Hastie, "Estimating the number of clusters in a dataset via the Gap statistic",J. Royal. Statis. Soc. B, 2000.

[8] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction", *Science*, vol. 290, pp. 2319-2323, 2000.

[9] S. Roweis, and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding", *Science* vol. 290, pp. 2323-2326, 2000.

[10] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*, MIT Press and McGraw-Hill, 2001.

[11] M. Belkin, and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering", *Advances in Neural Information Processing Systems*, vol. 14, pp. 585-591, 2002.

[12] A. Ben-Hur and I. Guyon. Detecting stable clusters using principal component analysis. In Methods in Molecular Biology, M.J. Brownstein and A. Khodursky (eds.) Humana press, pp. 159-182, 2003.

[13] S. Monti, P. Tamayo, J. Mesirov, T. Golub, "Consensus Clustering: A resampling-based method for class discovery and visualization of gene expression", Machine Learning Journal, 52 pp:91-118, 2003.

[14] J. Ham, D.D. Lee, S. Mika, and B. Schölkopf, "A Kernel View of the Dimensionality Reduction of Manifolds", *Proc. twenty-first int. conf. on Machine learning*, pp. 47-52, 2004.

[15] A. Baya, and P.M. Granitto, "ISOMAP based metrics for Clustering", *Inteligencia Artificial*, 37, pp 15–23, 2008.

[16] A. Baya, and P.M. Granitto, "Penalized K-Nearest-Neighbor-Graph Based Metrics for Clustering", *Submitted to Pattern Recognition*, 2008.