SciForum
Mol2Net

Iterative Kernel K-Means for Metagenomic Sequences

Isis Bonet [1,*], Andrea Mesa-Múnera [1], Adriana Escobar [1] and Juan Fernando Alzate [2]

[1] Escuela de Ingeniería de Antioquia, Envigado, Antioquia, Colombia; E-Mails: amesamu@gmail.com; adriana.escobarv@gmail.com

[2] Centro Nacional de Secuenciación Genómica, Facultad de Medicina, Universidad de Antioquia, Colombia; E-Mail: jfernando.alzate@udea.edu.co;

* E-Mail: ibonetc@gmail.com; Tel.: +57-4-3549090 (ext. 330)

Published: 4 December 2015

Abstract: This paper shows an iterative clustering method based on kernel *k*-means, which changes the parameter *k* automatically in each iteration of the algorithm. In addition, a way to initialize the centroids is proposed. The method is applied to a binning process in metagenomics using a complex database with different organisms. The aim of this method is to reduce the sensitivity of clusters based on strength measures. The results demonstrate that the proposed method is better than the simple kernel *k*-means for metagenome databases.

Keywords: Metagenomics; k-means; clustering; bioinformatics

1. Introduction

Metagenomics is the science that studies microbial DNA of many organisms recovered from environmental samples. Ever since the studies of DNA in a single organism the use of computational resources was an important need. Now this science has stirred the rise of new computational challenges. Next-generation sequencing technologies can sequence up to billions of bases in a single day at low cost, producing a huge amount of short fragments of DNA called reads. The next process and new challenge is to assemble these reads into longer sequences called contigs and scaffolds by a process of overlapping [1]. Assembling this huge amount of short reads was difficult in the classic genomic study for a single microorganism. Now assembling imposes great computational challenge because in metagenomics the data we are dealing with contains different microorganisms at uneven abundances.

Binning process for assignment of genomic fragments into taxonomic groups is one of the most important steps in the analysis of

metagenomic data, but in spite of several developed tools it is still a challenge for scientists. Similarity-based and supervised methods are more accurate than unsupervised methods because they are based on reference sequences, but for the same reason they are more time consuming and have limitations when they are dealing with unknown organisms or these are not present in their databases. The huge amount of reads or contigs to align with known sequences coupled with the big size of the known sequences databases are the cause of the high time consumption. Therefore if we reduce the amount of reads or contigs to align with, the time to find the sequence that match should considerably decrease. A previous clustering process can be an efficient way to provide different taxonomic groups in order to ease the analysis of a few fragments of sequences that probably belong to the same organism. This process can be used as a previous step in some processes in the study of metagenomic samples such as before the assembly or in the process of functional assignment. Some researchers have used variants of *k*-means [2,3], variants of Self-Organizing Maps [4], [5] and others clustering techniques [1,6]. In [7] a comparison of some different clustering methods is done.

The selection of an appropriate clustering method to represent the taxonomic groups is yet a challenge. The complexity and the high dimensional of the data are two of the problems to keep in mind in clustering metagenomics.

K-means is one of the most popular clustering algorithms, but it has some limitations. One of the most important disadvantages is the number of clusters needs to be specified by the user. A key limitation of *k*-means is the way to build the clusters, typically spherical clusters with similar size, which are linearly separable.

Taking into account the potential of k-means without forgetting its limitations this paper focuses on a kernel *k*-means method with a variant consisting of iterations in order to select an appropriate number *k* of clusters. Also a random way to select the centroids based on the distance between them is used improving the convergence of the method.

2. Data and Methods

2.1 Data

The database used in this paper was previously used by Bonet et al. [3]. It consists of assembled genomic sequences of different organisms: viruses, bacteria and eukaryotes from the FTP site of the Sanger institute.

Selected viral sequences include Influenza and Dengue virus genomes. Bacterial sequences come from Bacteroides dorei and Bifidobacterium longum. The selected eukaryotes included two fungi, one nematode and one insect.

The database contains 165014 contigs that ranged between 50 and 2962289 bases because the enormous difference in the size of contigs is needed to represent the sequences using biological or mathematical features.

2.2 Features

For the experiment some features were selected:

- GC: G + C content, that means the ratio between the number of G+C and the total of nucleotides of the sequence (A+T+G+C).
- Nucleotide frequencies: Number of occurrences of A, T, G and C in the sequence. It was normalized by the size of the sequence.
- Codon frequencies: Number of each possible codon in the sequence. It was normalized by the total of codons (64 codons)
- *k*-mer (*k*=4): are represented for the 256 possible tetranucleotides. It was compute as

the number of each tetranucleotide and normalized with the total of tetranucleotides in the sequence.

Features were used in all combinations, producing 15 databases.

2.3 Methods

K-means is one of the most popular clustering methods, despite the problem to estimate the parameter *k* (number of cluster). This algorithm finds a set of *k* centroids, and associates each instance in the data to the nearest centroid, based on a distance function [8].

Some researchers have focused on the initialization part of the method, based on the selection of better centroids in order to improve the convergence of the algorithm. One of the most known is *k*-means++ [9] and variants of it, including scalable *k*-means++ [10]. Most of these algorithms need to analyze the entire database, which requires a lot of time in large databases. Here we propose a simple and fast way to select a set of optimal centroids.

Appling *k*-means to massive data is easy because of its nature. Given a set of centroids, the assignment of each point to clusters can be done independently.

Kernel *k*-means works as k-means but applied in kernel space [11].

Here we proposed a clustering method based on kernel *k*-means.

Polynomial and cosine distance kernel were used to compare the sequences.

For the implementation of the clustering method, we used Weka [12], which is a free machine learning package that has implemented *k*-means.

3. Iterative kernel *k*-means

3.1 Selecting centroids

The process to select the centroids consists on:

1. Select *k* random points (*k* cases of the database).
2. Select a *k*+1 point. Compute the distance matrix of theses *k*+1 point. For each point, compute the average distance. Delete the point with lowest average.
3. Repeat step 2 until obtain an average greater than a threshold or a number of iterations.

Using this simple idea, we obtain a set of centroids more distant from each other, what is one of the objectives of the final clusters.

In this paper, we use *k* as the number of iterations for the step 2.

3.2 Iterative kernel *k*-means

The proposed process of clustering is based on the algorithm suggested by Bonet et al. [3] with the addition of a distance kernel. The distance kernel is based on a cosine transformation with a lineal kernel as is shown in equation 1.

$$CosineKernel(x_1, x_2) = \frac{k(x_1,x_2)}{\sqrt{k(x_1,x_1)*k(x_2,x_2)}} \quad (1)$$

where *k* is a kernel. In our case the linear kernel was use, i.e. $k(x_1, x_2)$ is a dot product of x_1 and x_2.

The process is following these steps:

Step 1: Select a tentative *k* (this *k* varies in the rest of the process), preferably a higher value than expected. Run *k*-means with the data using the initialization process and the cosine distance kernel described above.

Step 2: After getting the first set of clusters, they are evaluated based on measures of strengths of clusters. Clusters with low compactness, that is low distance inter-cluster, are used to build the new database to repeat the clustering process returning to step 2.

Step 3: Once the strengths measures are lower than a threshold, the last step is to minimize, if possible, the number of clusters. Clusters evaluation is repeated, for all clusters resulted of each iteration of *k*-means. Clusters with low separation between theirs centroids are merged into one.

Some different attributes are used to describe the sequences: GC content, nucleotides frequencies, codon frequencies and tetranucleotides. All combinations of features were tested, but the best performance was obtained using tetranucleotides.

Polynomial and cosine kernel were used for the kernel k-means algorithm. The best result was obtained with cosine kernel. The algorithm was tested with k between 5 and 15 achieving the best performance with k=15.

In metagenomics the aim to assign the sequences to a phylum is associated with the sensitivity taking into account the phylum that best represents each cluster. That means the sensitivity is measured centered around the percentage that represents each organism in each cluster. For this problem, we use the sensitivity of clusters to evaluate them.

4. Results and Discussion

A metagenome database composed of eight different organisms is used to evaluate the method.

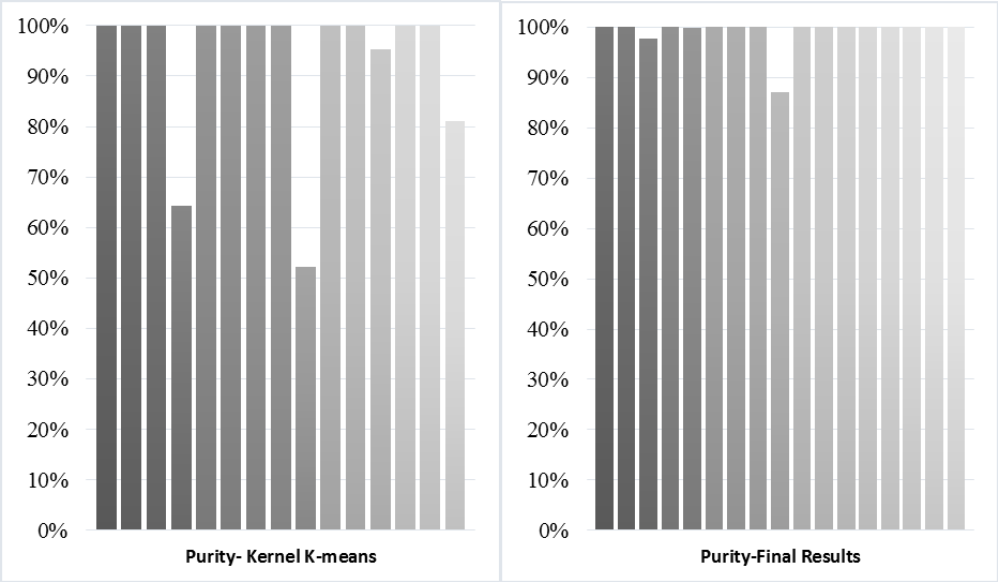

Figure 1. *K*-means vs. Iterative kernel *k*-means

Figure 1 shows the results with kernel cosine and k=15. The clusters obtained with kernel k-means (left) vs. the clusters obtained using the proposed algorithm with five iterations (right). The figure represents the percent of purity of the clusters that means, the percent of the genomic fragments that belongs to the predominant organism in the cluster.

The results of the last step of the model yielding a 99.1% of sensitivity of the clusters, which results are in the range of 87.14 and 100%. The error of misassigned sequences is 5.516%.

4. Conclusions

In this paper we present an algorithm based on kernel *k*-means. The algorithm was tested in a metagenome database. The result achieved by the proposed method in line with the objective of obtaining clusters with high sensitivity outperforms results obtained with a simple *k*-means. Taking into account the sensitivity of the clusters the model yielding a 99.1%.

Conflicts of Interest

The authors declare no conflict of interest

References

1. Reddy, R.M.; Mohammed, M.H.; Mande, S.S. Metacaa: A clustering-aided methodology for efficient assembly of metagenomic datasets. *Genomics* **2014**, *103*, 161-168.
2. Kelley, D.; Salzberg, S. Clustering metagenomic sequences with interpolated markov models. *BMC Bioinformatics* **2010**, *11*, 544.
3. Bonet, I.; Montoya, W.; Mesa-Múnera, A.; Alzate, J. Iterative clustering method for metagenomic sequences. In *Mining intelligence and knowledge exploration*, Prasath, R.; O'Reilly, P.; Kathirvalavakumar, T., Eds. Springer International Publishing: 2014; Vol. 8891, pp 145-154.
4. Weber, M.; Teeling, H.; Huang, S.; Waldmann, J.; Kassabgy, M.; Fuchs, B.M.; Klindworth, A.; Klockow, C.; Wichels, A.; Gerdts, G*., et al.* Practical application of self-organizing maps to interrelate biodiversity and functional data in ngs-based metagenomics. *The ISME journal* **2011**, *5*, 918-928.
5. Abe, T.; Kanaya, S.; Kinouchi, M.; Ichiba, Y.; Kozuki, T.; Ikemura, T. Informatics for unveiling hidden genome signatures. *Genome Research* **2003**, *13*, 693-702.
6. Kislyuk, A.; Bhatnagar, S.; Dushoff, J.; Weitz, J. Unsupervised statistical clustering of environmental shotgun sequences. *BMC Bioinformatics* **2009**, *10*, 316.
7. Li, W.; Fu, L.; Niu, B.; Wu, S.; Wooley, J. Ultrafast clustering algorithms for metagenomic sequence analysis. *Briefings in Bioinformatics* **2012**, *13*, 656-668.
8. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, University of California Press: Berkeley, Calif., 1967; pp 281-297.
9. Arthur, D.; Vassilvitskii, S. In *K-means ++: The advantages of careful seeding*, 8th Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, 7-9 January 2007, 2007; New Orleans, pp 1027-1035.
10. Bahmani, B.; Moseley, B.; Vattani, A.; Kumar, R.; Vassilvitskii, S. Scalable k-means++. *Proc. VLDB Endow.* **2012**, *5*, 622-633.
11. Scholkopf, B.; Smola, A.; Muller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* **1998**, *10*, 1299-1319.

12. Witten, I.; Frank, E. *Data mining: Practical machine learning tools and techniques*. 2nd ed.; Morgan Kaufmann: San Francisco, 2005; p 525.