Unsupervised Binning of Metagenomic Assembled Contigs Using Improved Fuzzy C-Means Method

Yun Liu, Tao Hou, Bing Kang, and Fu Liu

Abstract—Metagenomic contigs binning is a necessary step of metagenome analysis. After assembly, the number of contigs belonging to different genomes is usually unequal. So a metagenomic contigs dataset is a kind of imbalanced dataset and traditional fuzzy c-means method (FCM) fails to handle it very well. In this paper, we will introduce an improved version of fuzzy c-means method (IFCM) into metagenomic contigs binning. First, tetranucleotide frequencies are calculated for every contig. Second, the number of bins is roughly estimated by the distribution of genome lengths of a complete set of non-draft sequenced microbial genomes from NCBI. Then, IFCM is used to cluster DNA contigs with the estimated result. Finally, a clustering validity function is utilized to determine the binning result. We tested this method on a synthetic and two real datasets and experimental results have showed the effectiveness of this method compared with other tools.

Index Terms—Metagenomic binning, k-mer, clustering method, FCM

1 Introduction

Next-generation sequencing technologies are applied to metagenome research to sequence the entire community of microbial species, including culturable and unculturable species [1], [2], [3]. Existing metagenomic projects, e.g., Acid Mine Drainage Biofilm (AMD) project [4], Human Gut Microbiome (HGM) project [5] and gut microbiome in obese and lean twins [6], have made profound discoveries about microbial communities.

The task of binning metagenomic contigs into groups is a necessary step of metagenome analysis and remains a significant challenge [7]. Existing binning methods for metagenomic datasets fall into two categories, supervised and unsupervised binning methods [2], [8]. Supervised binning methods are nearly all based on reference sequences or pre-computed models [9], [10], [11]. However, those methods will be ineffective if there is a lack of reference information [12]. To solve this problem, binning methods with unsupervised techniques for metagenomic datasets containing unknown species, have been developed in recent years, e.g., AbundanceBin [13], MetaCluster 3.0 [14], GroopM [7], TF-ESOM [15] and others.

Recently, new binning methods have emerged that use differential coverage patterns across multiple related metagenomes, e.g., GroopM [7], CONCOCT [16] and the approach described by Nielsen et al. [17]. For a given ecosystem, a series of related metagenomes, spatial or temporal, are sequenced. Then reads of multiple metagenomes are co-assembled into contigs by an assembler and coverage profiles can be obtained simultaneously. Methods of this type are based on the assumption that contigs with similar coverage profiles are probably originated from the same microbial population, and the methods have shown great promise for improving binning fidelity by combining coverage profiles and composition-based methods [7]. In this paper, we will include one of these methods, GroopM, for comparison.

After assembly, for a number of reasons, e.g., genome size, relative abundance in a community and so on, the number of contigs belonging to different genomes is usually unequal. So metagenomic contigs dataset is a kind of imbalanced dataset, in which the number of samples inter-class is usually unequal. Classes with relatively small number of samples are called minority classes, whereas classes that contain relatively large number of samples are called majority classes. Traditional fuzzy c-means method (FCM) has a poor performance on this type of dataset [18]. In this paper, we will introduce an improved version of fuzzy c-means method (IFCM) [19] for unsupervised binning of metagenomic contigs. Cluster size is considered as IFCM progresses, and is added to the computation formula of the membership matrix. This improvement addresses the defect of FCM, which tends to cluster data into groups with similar size [18], [20]. However, IFCM does not have the ability to determine the number of clusters. A common method is to implement FCM several times with a series of the number of clusters, and then select the best clustering structure according to a cluster validation test [21]. In this paper, we use the validity function $FS(c)$, proposed by Fukuyama and Sugeno [22] to choose the best number of clusters. By combining IFCM and $FS(c)$, the method in this paper could effectively handle the metagenomic dataset and output the number of clusters automatically as well. We tested the binning performance of IFCM on a synthetic dataset and two real datasets. Experimental results have showed that this method has better performance

- *The authors are with the College of Communication Engineering, Jilin University, Changchun 130000, China. E-mail: laoniu313@qq.com, ht_happy@126.com, {kangbing, liufu}@jlu.edu.cn.*

Manuscript received 28 Apr. 2016; accepted 20 May 2016. Date of publication 7 June 2016; date of current version 6 Dec. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2016.2576452

than traditional FCM, MetaCluster3.0, and TF-ESOM and works as good as GroopM for synthetic dataset and outperforms it for real datasets.

2 Materials and Methods

2.1 Construction of the Feature Matrix by K-mer Frequencies

K-mer is a substring of DNA sequence with length k, so there are 4^k kinds of k-mers in it. For a DNA contig with length l, there are a total of $(l-k+1)$ k-mers. Therefore, a DNA sequence can be represented by a feature vector $f = [f_1, f_2, \ldots, f_{4^k}]$ which should meet the following condition:

$$\sum_{i=1}^{4^k} f_i = l - k + 1, \tag{1}$$

where f_i is the occurrence number of ith mer in a DNA contig. In this paper, we set $k=4$ as it is often used in metagenomic binning [23], [24] and Zhou et al. has proved that $k=4$ is a good choice when barcoding a genome with DNA fragment size from 1,000 nt to 10,000 nt [25]. As each DNA fragment can be sequenced from each strand of DNA genome, the frequency of one k-mer and its reverse complement k-mer can be combined to be one, and this will reduce the dimension of the feature vector f nearly by half. Considering the palindromic sequence, the dimension of f will be $(4^k + 4^{k/2})/2$ if k is even. So, the feature vector will be 136-dimensional.

For a metagenomic dataset that contains N DNA contigs, the feature matrix $F_{136\times N}$ is constructed, where f_{ij} represents the occurrence number of ith mer in jth contig. Then F is normalized as follows:

$$x_{ij} = f_{ij} \Bigg/ \sum_{i=1}^{136} f_{ij}. \tag{2}$$

2.2 Estimation of Number of Species

The number of clusters is often a necessary initial condition for unsupervised classification methods [26], [27]. In this part, a rough estimate of the number of species that assembled well is being sought according to the size of assembled contigs, average coverage of metagenome and distribution of complete genome length.

In a metagenomic dataset containing c species, the total length of all genomes in community G is [28]

$$G = \sum_{i=1}^{c} \eta_i G_i, \tag{3}$$

where G_i is the genome length of ith species, η_i is the coverage of G_i. Suppose that $\bar{\eta}$ is the average coverage of c species. Then we could get the following relationship:

$$c \cdot \bar{\eta} G_{\min} \leq G \leq c \cdot \bar{\eta} G_{\max}, \tag{4}$$

where $G_{\min}$ and $G_{\max}$ are the minimum and maximum genome lengths in a metagenome. According to this relationship, the number of species is in the following interval:

$$\frac{G}{\bar{\eta} G_{\max}} \leq c \leq \frac{G}{\bar{\eta} G_{\min}}. \tag{5}$$

To estimate the number of species in assembled contigs, we need to assess $\bar{\eta}$, $G_{\min}$ and $G_{\max}$ of a metagenome. First, we use Nonpareil [29], a redundancy based method that does not rely on high-quality assemblies, operational taxonomic unit calling or comprehensive reference databases, to assess $\bar{\eta}$ of a metagenome. For multiple related metagenomes, the average $\bar{\eta}$ of metagenomes is utilized.

Different species have different genome sizes, and genomes in a metagenome are unknown in advance. To determine $G_{\min}$ and $G_{\max}$, a complete set of non-draft sequenced microbial genomes (including 2,573 bacteria) (Supp_complete_genomes.xls) were downloaded from the National Center for Biotechnology Information (NCBI) website (ftp://ftp.ncbi.nih.gov/genomes/Bacteria/) on October 21st, 2015. The genomes were filtered to remove plasmids to allow for an analysis of only chromosomal DNA. Supplementary file, Supp_complete_genomes.xls, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2016.2576452, lists the information of the 2,573 bacteria genomes, in which the bacteria name, accession number and genome length are included. In these genomes, the longest genome reaches 1.3×10^7 bps, while the shortest is only 1.3×10^5 bps, which is almost 100 times shorter than the longest one. For computational efficiency, we do not need to select this two values for $G_{\min}$ and $G_{\max}$. In this paper, we sort these genomes according to their lengths in ascending sequence and use the lengths of 129th and 2,444th genomes for $G_{\min}$ and $G_{\max}$, which are the 5 and 95 percent of these genomes respectively. The results are $G_{\min} = 9.4 \times 10^5$bps and $G_{\max} = 6.4 \times 10^6$bps. So the interval, $[G_{\min}, G_{\max}]$, could include 90 percent of these genomes.

Finally, for a metagenomic dataset, the value range of the number of species $[c_{\min}, c_{\max}]$ could be determined by

$$c_{\min} = \frac{N \times l}{G_{\max} \times \bar{\eta}} \tag{6}$$

and

$$c_{\max} = \frac{N \times l}{G_{\min} \times \bar{\eta}}, \tag{7}$$

while N and l are the number and average length of DNA contigs respectively.

In this paper, we use the interval of species number as the initial number of clusters for IFCM, described in next section. It is worth noticing that some species could not be distinguished by tetramer frequencies, so the number of species is not the same as the number of clusters. Our purpose, however, is not to estimate the species number very accurately, but to determine a relatively larger interval as the initial clusters number for IFCM.

2.3 Cluster Progress Using IFCM

2.3.1 Brief Introduction to FCM

For a metagenomic dataset with N DNA contigs, suppose that $X = \{x_{ij}\}_{136\times N}$ is the normalized feature matrix computed by (2). FCM partitions these contigs into c clusters through an iterative minimization process of an objective function $J(\boldsymbol{U},\boldsymbol{V})$ [30], which is defined as

$$J(\boldsymbol{U},\boldsymbol{V})=\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{q}d(\boldsymbol{x}_i,\theta_j), \tag{8}$$

where $\boldsymbol{x}_i$ is the ith contig of $\boldsymbol{X}$, θ_j is the jth cluster center, $u_{ij}\in[0,1]$ is the membership value of $\boldsymbol{x}_i$ to θ_j with a constraint $\sum_{j=1}^{c}u_{ij}=1,\ q\in[1,+\infty)$ is the fuzziness degree, and $d(\cdot)$ is the similarity measure. In this paper, Euclidean distance is utilized with this process.

Utilizing Lagrange Multiplier, the objective function $J(\boldsymbol{U},\boldsymbol{V})$ is minimized and the membership matrix $\boldsymbol{U}$ would be

$$u_{rs}=\frac{1}{\sum_{j=1}^{c}\left(d(\boldsymbol{x}_r,\theta_s)/d(\boldsymbol{x}_r,\theta_j)\right)^{2/(q-1)}},r=1,2,\ldots,N,s=1,2,\ldots,c. \tag{9}$$

The cluster center is

$$\theta_j=\frac{\sum_{i=1}^{N}u_{ij}^{q}\boldsymbol{x}_i}{\sum_{i=1}^{N}u_{ij}^{q}},\quad 1\le j\le c. \tag{10}$$

So the FCM algorithm can be summarized as below:

1) Construct membership matrix $\boldsymbol{U}$ with random decimal fraction.
2) Compute cluster centers using (10).
3) Update $\boldsymbol{U}$ using (9).
4) Repeat step (2) to (3) until $\|\boldsymbol{U}^{(t)}-\boldsymbol{U}^{(t-1)}\|<\varepsilon$, where ε is a very small number.
5) Defuzzification. Assign every contig to the cluster with the highest membership value.

2.3.2 Clustering Contigs Using IFCM

FCM cannot deal with an unbalanced dataset very well and IFCM is designed to tackle this defect, by taking into account the size of each cluster in every iteration process. The size of a cluster means the number of contigs in this cluster.

The size of cluster is defined as

$$f_j=\sum_{i=1}^{N}u_{ij}\quad ,j=1,2,\ldots,c. \tag{11}$$

Then (9) can be rewritten as

$$u_{rs}=\frac{f_s\Big/(d(\boldsymbol{x}_r,\theta_s))^{2/(q-1)}}{\sum_{j=1}^{c}f_j\Big/\left(d(\boldsymbol{x}_r,\theta_j)\right)^{2/(q-1)}},r=1,2,\ldots,N,s=1,2,\ldots,c. \tag{12}$$

FCM tends to partition samples into clusters with similar size, so some samples from the majority class will be clustered into minority clusters incorrectly. In (12), membership values of each sample are codetermined by distance to every cluster center and cluster size. By doing this, samples from the majority class but near the classification boundary will not be clustered to a minority class. Thus, IFCM could improve the performance of FCM for unbalanced datasets.

Accordingly, the objective function will be

$$J(\boldsymbol{U},\boldsymbol{V})=\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{q}d(\boldsymbol{x}_i,\theta_j)\Big/f_j. \tag{13}$$

So the pipeline of IFCM can be described as below:

1) Construct membership matrix $\boldsymbol{U}$ with random decimal fraction.
2) Compute cluster size using (11).
3) Compute cluster centers using (10).
4) Update $\boldsymbol{U}$ using (12).
5) Repeat steps (2) to (4) until $\|\boldsymbol{U}^{(t)}-\boldsymbol{U}^{(t-1)}\|<\varepsilon$, where ε is a very small number.
6) Defuzzification. Assign every contig to the cluster with the highest membership value.

In this paper, IFCM algorithm is implemented $c_{\max}-c_{\min}+1$ times with different initial number of clusters belonging to $[c_{\min},c_{\max}]$.

2.4 Choose the Best Clustering Result Using FS(c)

After the clustering progress, we have obtained $c_{\max}-c_{\min}+1$ clustering results with different numbers of clusters. In this section, $FS(c)$ is utilized to select the best one from the $c_{\max}-c_{\min}+1$ clustering results. $FS(c)$ is defined by Fukuyama and Sugeno [22]:

$$FS(c)=\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{q}d\big(\boldsymbol{x}_i,\theta_j\big)-\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{q}d\big(\theta_j,\overline{\theta}\big), \tag{14}$$

where $\overline{\theta}=\sum_{j=1}^{c}\theta_j/c$. $FS(c)$ is a classical validity function for fuzzy clustering method, which takes into account fuzzy memberships and data structure simultaneously [31]. $FS(c)$ is utilized in this way to choose the best clustering result in this paper.

In (14), the first term is the objective function of FCM (see (8)), which should be as small as possible. The second term is to measure the dispersion degree of cluster centers to average center $\overline{\theta}$. Higher distance among clusters is expected when clustering a dataset, so the value of this term should be as large as possible. Thus, the best number of clusters is:

$$c^*=\min_{c_{\min}\le c\le c_{\max}}FS(c). \tag{15}$$

2.5 Datasets

2.5.1 Synthetic Dataset

The synthetic dataset used for the binning experiment in this paper is from Imelfort [7], and can be downloaded from https://github.com/minillinim/GroopM_test_data. The operational taxonomic unit table was generated from three related soil samples of Stordalen mire [32] and contains 1,159 unique OTUs [7]. Error-free 100 bp paired-end reads were generated from fully sequenced and permanent draft genomes (reference sequences), which were downloaded from the Integrated Microbial Genomes (IMG) database at http://img.jgi.doe.gov. Finally, the synthetic reads were co-assembled using Velvet version 1.2.07 [33] and contigs shorter than 500 bp were removed, leaving 5,668 contigs. To verify bin assignments for each contig, BLAST version 2.2.25+ [34] is utilized to align contigs to 1,159 IMG reference genomes. It was observed that a number of contigs mapped equally well to multiple closely related reference genomes. These genomes are from two groups, five genomes of genus *Thermotoga* and two strains of the species *Oenococcus oeni*. Then the five *Thermotoga* variants and two *Oenococcus* were treated as single populations. Finally, a total of 259 genomes were identified with the best match for at least one contig.

In this dataset, there are 5,668 contigs from 259 genomes, with an average number of 24 contigs in each genome. There are 41 genomes that contain more contigs than the average number, and contigs coming from these genomes are called majority contigs in this paper. The total number of majority contigs is 4,926. The rest 218 genomes only contain 742 contigs, especially there are 100 genomes that only contain 1 contig. These 742 contigs are called minority contigs. The detailed list of composition genomes and number of contigs of each genome are available in Supplementary Material (Supp_synthetic_dataset_information.xls, available online).

Using Nonpareil [29], the average coverage of this dataset is 0.94.

2.5.2 Real Datasets

Qin et al. [5] collected metagenomic samples from feces of 124 European adults. SOAPdenovo [35] was used to assemble DNA reads into contigs. We selected contigs from two of those samples, MH0002 and MH0012, (ftp://public.genomics.org.cn/BGI/gutmeta/Single_Sample_contig), as the real metagenomic datasets. MH0002 and MH0012 both have three related metagenomes, making it possible to comparisons with GroopM, which is a recently emerged coverage-based binning method. There are 53,507 and 140,992 contigs in MH0002 and MH0012 respectively. Contigs less than 500bp were removed. Then, BLASTn tool (http://blast.ncbi.nlm.nih.gov/Blast.cgi) was utilized to verify the assignment for each contig. When using BLASTn, 'Reference genomic sequences (refseq_genomic)' was selected as the database and other settings were default. It is possible that some contigs come from genomes that are not included in RefSeq, and those will be assigned to one of the closest relatives in RefSeq. To eliminate this influence, contigs with less than 90 percent query cover were removed and 45,201 and 41,786 contigs were left for binning (see Supplementary material, Supp_real_dataset_information.xls, available online). For chimeric contigs, that are mapped equally well to multiple closed related genomes, we used the processing method for synthetic dataset and treated these genomes as single populations. Finally, in MH0002 a total of 244 genomes were identified with the best match for at least one contig. In MH0012 this number is 246. The minimum length of contigs in this two datasets is 500bp. Using Nonpareil [29], the average coverage of MH0002 and MH0012 is 0.67 and 0.73 respectively.

The real datasets, titled real1.fna and real2.fna, along with the source code of IFCM, are available in GitHub (https://github.com/liuyun313/IFCM).

Contigs of synthetic and real datasets for all compared methods were processed in exactly the same way described above.

3 Results and Discussion

In this section, we will evaluate the IFCM on a synthetic and two real datasets, and compare its performance with FCM, MetaCluster3.0, TF-ESOM and GroopM. TF-ESOM [15] and GroopM [7] are two main recent binning tools. The series of MetaClusters are also commonly used unsupervised binning tools (http://i.cs.hku.hk/~alse/MetaCluster/index.html). However, higher versions than 3.0 are designed for pair-end reads, so we chose MetaCluster3.0 for comparison. FCM, TF-ESOM, MetaCluster3.0 and IFCM have the same data points for clustering. GroopM, however, is different, and it uses co-varying coverage profiles across multiple related metagenomes for contigs binning.

3.1 Experimental Results

To evaluate the binning performance of proposed method, we mainly focus on the number of contigs that are clustered correctly, as well as the corresponding number of base pair. Moreover, the number of outputted clusters/bins is also an important criterion. As far as we know, many unsupervised binning methods, i.e. GroopM and MetaClusters, also use these criteria for evaluation.

The binning performances of IFCM and other comparative methods are evaluated at the level of species. In a cluster, it is possible that contigs are from different genomes. One genome, that contains the maximum number of contigs in a cluster, is determined as the dominant genome in this cluster, and its contigs are the correctly clustered contigs in this cluster, and the number of base pairs of these contigs are the correctly clustered base pairs. All other contigs in this cluster are correctly clustered contigs. Dominant genomes of all clusters are determined by this way, as well as the correctly clustered contigs and base pairs. All methods in this paper are evaluated in the same way.

3.1.1 Experimental Results with a Synthetic Dataset

Table 1 shows the binning results of traditional FCM, GroopM, TF-ESOM, MetaCluster 3.0 and IFCM, in which the number of bins, correctly clustered number of minority, majority and total contigs and the corresponding number of assembled bases, are listed. The binning results of GroopM and TF-ESOM are obtained from the supplementary material of Imelfort et al. [7].

Compared with traditionnal FCM, IFCM achieved a better performance as IFCM correctly clustered much more both minority contigs and majority contigs, as well as the corresponding bases, and outputted 18 more bins, illustrating that the improvement of IFCM is effective. MetaCluster 3.0 only outputted 4 bins and correctly clustered 636 contigs, so the ability of MetaCluster 3.0 for binning contigs need much more improvement. GroopM and IFCM achieved an overall better performance than TF-ESOM, both for the number of bins and for the number of correctly clustered contigs. As GroopM uses coverage information for binning, it achieved a very good performance for this dataset and correctly clustered more majority contigs and bases but less minority contigs and bases than IFCM. We can get that IFCM has a similar performance with GroopM, which is worth mentioning as IFCM only utilizes tetranucleotide features. Overall, IFCM correctly clustered the

TABLE 1
Binning Performance of Synthetic Dataset

	No. of bins	Genomes		Total
		Contigs ratio *<0.0039	Contigs ratio>0.0039	
		Correctly clustered	Correctly clustered	
FCM	31	15 24,664 bp	1869 26,183,184 bp	1884 26,207,848 bp
GroopM	53	62 14,340,182 bp	4449 116,142,136 bp	4511 130,482,318 bp
TF-ESOM	45	48 13,974,325 bp	2557 99,300,433 bp	2605 113,274,758 bp
MetaCluster3.0	4	–	636 16,053,721 bp	636 16,053,721 bp
IFCM	48	116 14,920,644 bp	4224 106,364,534 bp	4340 121,285,178 bp

Contigs ratio of species is the proportion of contigs from the same species to total number of contigs. For this dataset, which contains 259 verified bins, we set the genomes'contigs ratios that are less than the average ratio (1/259 = 0.0039), as the minority classes. Contigs belong to minority class are called minority contigs. Majority contigs are defined in similar way.

maximum number of minority contigs and corresponding bases, so IFCM has a good prospect of binning minority contigs of metagenomic dataset.

More detailed results are listed in Supplementary Material (Supp_binning_result1.xls, available online). This supplementary file contains contigs assignments of the five methods, genome name and correctly clustered contigs number of every bin, number of minority and majority contigs that are clustered correctly, and so on.

Overall, IFCM and GroopM produced very similar population genome bins (Fig. 1). Results of IFCM and GroopM have 41 same genome bins. The additional 12 bins of GroopM contain 7 minority bins and 5 majority bins, and GroopM correctly clustered 98 contigs in these bins. The additional 7 bins of IFCM include 6 minority bins and 1 majority bin, which are less than that of GroopM, however IFCM correctly clustered 132 contigs in these bins that is 34 more than GroopM.

3.1.2 Experimental Results with Real Datasets

We also tested the binning performance of IFCM on two real datasets. Using TF-ESOM, contigs shorter than 2Kbp will be excluded for binning [15]. In real datasets, the average lengths MH0002 and MH0012 are 1,253bp and 1,553bp respectively. So, we compared IFCM with GroopM and MetaCluster 3.0.

a) MH0002. The binning results of IFCM, GroopM and MetaCluster3.0 on MH0002 are listed in Table 2. Because the number of contigs in this dataset is much more than that of synthetic dataset, the performance of the three methods all become worse. Similar to binning results of synthetic dataset, IFCM correctly clustered the maximum number of minority contigs and corresponding bases, 1,702 more contigs than MetaCluster 3.0 and 4,156 more than GroopM. For majority contigs, IFCM correctly clustered 5,182 more than MetaCluster3.0 and 1,890 more than GroopM, which is different from the results in synthetic dataset. Overall, IFCM

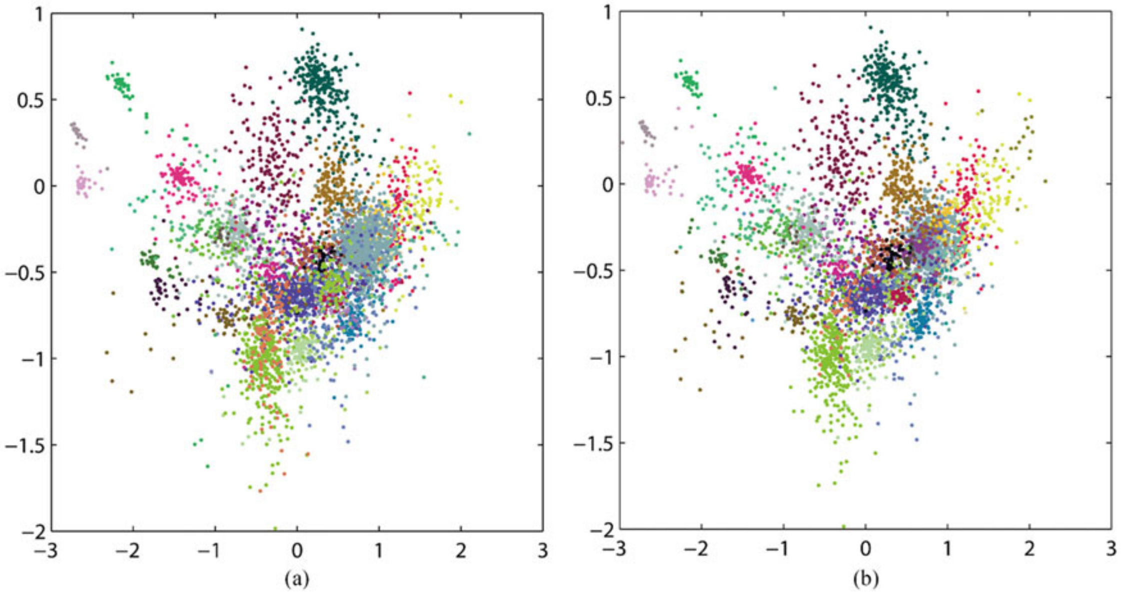


Fig. 1. A comparison of distribution of correctly assigned contigs of GroopM and IFCM on synthetic dataset. Contigs are positioned a ccording to the first two principal components of their tetranucleotide frequencies. Contigs clustered into one bin are of same color. (a) and (b) plot contigs assignments of GroopM and IFCM, respectively.

TABLE 2
Binning Results of IFCM, GroopM, and MetaCluster3.0 on mh0002

	No. of bins	Species		Total
		Contigs number<1051*	Contigs number>1051	
		Correctly clustered	Correctly clustered	
MetaCluster3.0	8	2,868 4,904,636 bp	7,690 10,153,132 bp	10,558 15,057,768 bp
GroopM	32	414 577,166 bp	10,982 14,964,262 bp	11,396 15,541,428 bp
IFCM	31	4,570 7,679,840 bp	12,872 18,355,571 bp	17,442 26,035,411 bp

**The maximum number of contigs belonging to one genome is 2102, while the minimum number is 1. So we set the average value, 1051, as the boundary of minority classes and majority classes.*

correctly clustered 17,442 contigs, while this number of MetaCluster3.0 and GroopM are 10,558 and 11,396 respectively, which are both less than that of IFCM. As for the correctly clustered base pairs, IFCM clustered the maximum number, almost 10 Mbps more than GroopM and MetaCluster3.0. So we can get that IFCM achieved a better performance than MetaCluster 3.0 and GroopM in this experiment. More detailed binning results are in Supplementary Material (Supp_binning_result2.xls, available online).

To visualize the binning results of the three methods in this experiment, we plot contigs in MH0002 into a 3D space according to the first three principal components of tetranucleotide frequencies (Fig. 2). Fig. 2a plots verified bins with different and random colors, and is seems to be somewhat confused, the reason of which is that the first three principal components of tetranucleotide frequencies only contribute 40 percent of all the features. From Fig. 2b, 2c, and 2d, we can get that IFCM correctly clustered much more contigs than GroopM and MetaCluster 3.0, and the distribution of these contigs is more wide than the above two methods.

It is worth noting that GroopM achieves a similar performance with IFCM for synthetic dataset, but not better performance than IFCM for MH0002. The reason is maybe the coverage of synthetic dataset and MH0002. By using Nonpareil, the average coverage of synthetic dataset is up to 0.94, while this value of MH0002 is only 0.67. GroopM uses differential coverage to bin assembled contigs from related metagenomes. So whether GroopM, as well as other similar methods that use differential coverage for binning, is sensitive to the coverage of metagenome, it is necessary to investigate further.

b) MH0012. The binning results of IFCM, GroopM and MetaCluster3.0 on MH0012 are listed in Table 3. Similar to binning results of MH0002, IFCM correctly clustered the maximum number of minority contigs, 3,168 and 2,100 more than MetaCluster 3.0 and GroopM respectively. For majority contigs, IFCM correctly clustered 3,687 and 2,560 more than MetaCluster 3.0 and GroopM respectively. As for the correctly clustered base pairs, IFCM achieved the maximum number, 10 Mbps more than the other two methods. The more detailed binning results of this experiment are listed in Supplementary material (Supp_binning_result3.xls, available online). So we can get that IFCM still achieved a better performance than MetaCluster 3.0 and GroopM in this experiment.

The coverage of this dataset is 0.73 by Nonpareil, which is much lower than that of synthetic dataset used in this paper. Same to the binning results of MH0002, GroopM achieved a not better performance than IFCM in this experiment. So it is very possible that GroopM, as well as other similar methods, is sensitive to the coverage of metagenomes, and lower coverage may affect its binning performance. However, we cannot draw this conclusion without rigorous analysis and verification and this is one of our future research objectives.

3.2 Running Time and Memory Usage

We run IFCM on a 64-bit Linux environment with a 16 GB memory installed. The CPU time and peak memory usage are listed in Table 4.

4 Conclusion

In this paper, IFCM for metagenomic contigs binning is introduced. As an improved version of FCM, IFCM could achieve better performance for unbalanced datasets. Binning experiments on one synthetic and two real datasets were conducted. Experimental results have showed the effectiveness of IFCM compared with FCM, TF-ESOM, MetaCluster3.0 and GroopM.

We also find that GroopM achieved a similar performance with IFCM on synthetic dataset, but the performances on real datasets are not better than IFCM. By using Nonpareil [29], a redundancy based coverage assessing method that does not rely on high-quality assemblies, operational taxonomic unit (OTU) calling or comprehensive reference databases, we find that the coverage of synthetic dataset is up to 0.94, while this values of real datasets are only 0.67 and 0.73. The reason of not better performance of GroopM for real datasets than synthetic dataset is maybe due to the relatively low coverage of real datasets. However, further and rigorous analysis and investigations are necessary to draw this conclusion, which is one of our future research objectives. IFCM only uses tetranucleotide frequencies for binning, making it suitable for metagenomes with high or low coverages.

Current binning methods for metagenomic datasets are mature but still have space for improvement. There are two main problems for unsupervised binning method that need to be improved. The first one is the relatively small number of correctly clustered contigs. In synthetic dataset used in

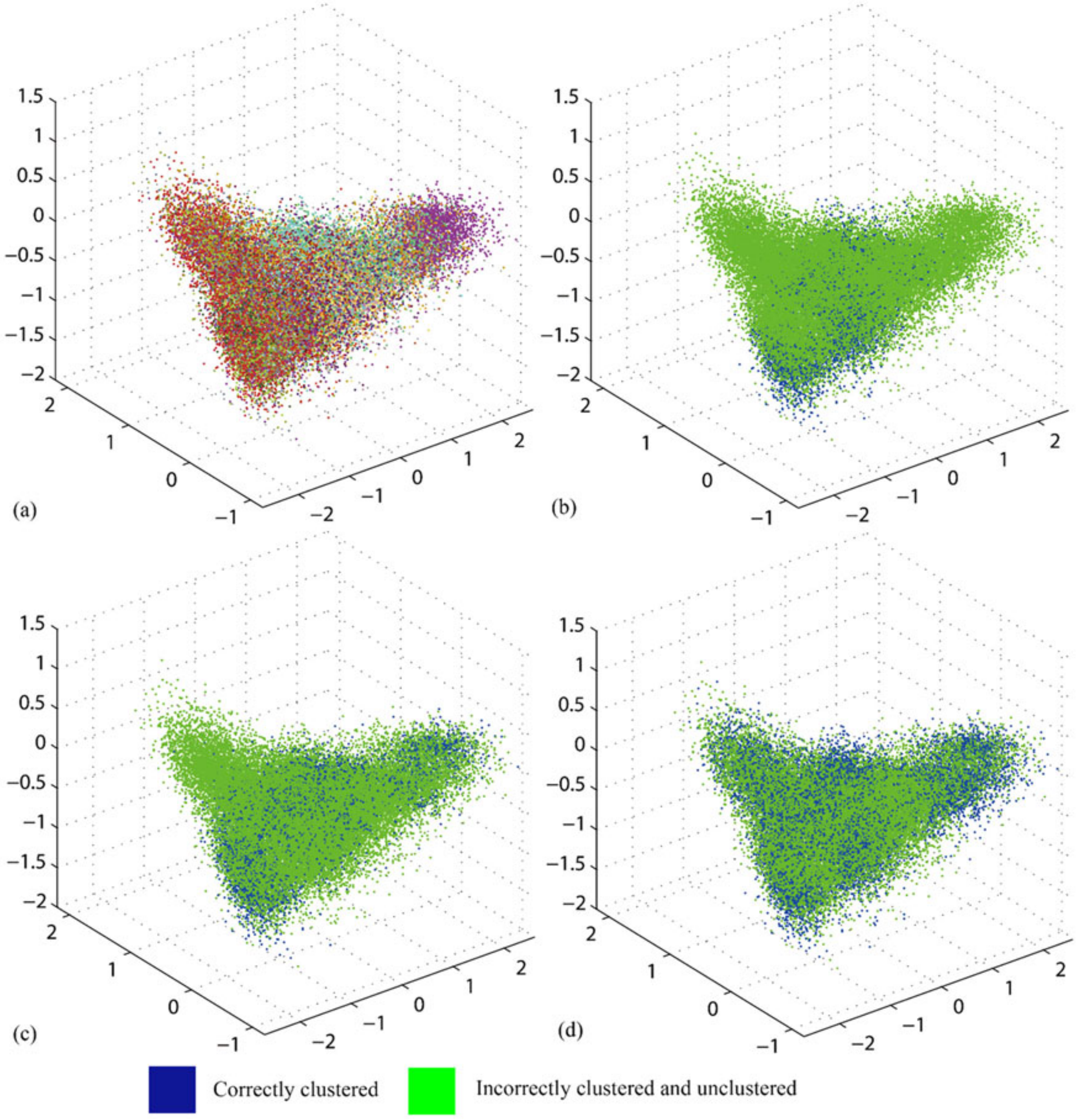


Fig. 2. The distribution of tetranucleotide frequencies and bin assignments of MetaCluster 3.0, GroopM, and IFCM for real metagenomic contigs. Contigs are positioned according to the first three principal components of tetranucleotide frequencies. (a) shows the verified bins with different and random colors. (b),(c), and (d) are binning results of MetaCluster 3.0, GroopM, and IFCM, respectively, in which correctly clustered contigs are colored in blue, whereas incorrectly clustered and unclustered contigs are colored in green.

this paper, there are only 5,668 contigs and GroopM and IFCM could clustered most of these contigs correctly. But for the real datasets, MH0002 and MH0012, after filtering there are 45,201 and 41,786 contigs respectively. IFCM achieved the best performance for this two datasets compared with GroopM and MetaCluster3.0, but it only correctly clustered 17,442 and 15,223 contigs respectively. The second one is the small number of discovered species,

TABLE 3
Binning Results of IFCM, GroopM, and MetaCluster3.0 on mh0012

	No. of bins	Species		Total
		Contigs number<1051*	Contigs number>1051	
		Correctly clustered	Correctly clustered	
MetaCluster3.0	7	984 1,624,970 bp	7,384 8,066,196 bp	8,368 9,691,166 bp
GroopM	25	2,052 3,305,892 bp	8,511 6,908,311 bp	10,563 10,214,203 bp
IFCM	26	4,152 10,454,721 bp	11,071 12,775,713 bp	15,223 23,230,434 bp

**The same boundary of minority and majority classes was used for MH0012, because MH0002 and MH0012 have the similar number of contigs and genomes.*

TABLE 4
CPU Time and Memory Usage of IFCM

Datasets	CPU time	Peak memory usage
Synthetic	2,854s	115 M
MH0002	3,291s	876 M
MH0012	3,301s	823 M

which exists both in synthetic dataset and real datasets. The three datasets used in this paper all contain more than 200 species, while GroopM and IFCM only outputted dozens of bins. So in our future research, we will focus on this two drawbacks and design more powerful unsupervised binning tools.

Availability: We have created a repository on GitHub for the source code, and it can be downloaded from https://github.com/liuyun313/IFCM. The two real datasets are also in this repository. The synthetic dataset can be downloaded from https://github.com/minillinim/GroopM_test_data.

Acknowledgments

The authors thank Mingyu Guo, Xuecheng An, and Shoukun Jiang for providing programming support. They are also grateful to the authors of MetaClusters and GroopM for providing the tools and dataset so that we could do comparative experiments in this paper. This work is financially supported by the National Natural Science Foundation of China (NO. 61503151) and Science, Technology development project of Jilin Province (NO. 20140204046GX) and Natural Science Foundation of Jilin Province (No. 20160520100JH). Fu Liu is the corresponding author. They also want to thank the two anonymous reviewers for their useful input.

References

[1] J. Droge and A. C. McHardy, "Taxonomic binning of metagenome samples generated by next-generation sequencing technologies," *Brief Bioinf.*, vol. 13, no. 6, pp. 646–655, Nov. 2012.

[2] S. S. Mande, M. H. Mohammed, and T. S. Ghosh, "Classification of metagenomic sequences: Methods and challenges," *Briefings Bioinf.*, vol. 13, no. 6, pp. 669–681, Nov. 2012.

[3] Y. Wang, H. C. Leung, S. M. Yiu, and F. Y. Chin, "MetaCluster 5.0: A two-round binning approach for metagenomic data for low-abundance species in a noisy sample," *Bioinf.*, vol. 28, no. 18, pp. i356–i362, 2012.

[4] G. W. Tyson, et al., "Community structure and metabolism through reconstruction of microbial genomes from the environment," *Nature*, vol. 428, no. 6978, pp. 37–43, 2004.

[5] J. Qin, et al., "A human gut microbial gene catalogue established by metagenomic sequencing," *Nature*, vol. 464, no. 7285, pp. 59–65, 2010.

[6] P. J. Turnbaugh, et al., "A core gut microbiome in obese and lean twins," *Nature*, vol. 457, no. 7228, pp. 480–484, 2009.

[7] M. Imelfort, D. Parks, B. J. Woodcroft, P. Dennis, P. Hugenholtz, and G. W. Tyson, "GroopM: An automated tool for the recovery of population genomes from related metagenomes," *J. Peer*, vol. 2, 2014, Art. no. e603.

[8] R. Liao, R. Zhang, J. Guan, and S. Zhou, "A new unsupervised binning approach for metagenomic sequences based on N-grams and automatic feature weighting," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 1, pp. 42–54, Jan./Feb. 2014.

[9] D. H. Huson, A. F. Auch, J. Qi, and S. C. Schuster, "MEGAN analysis of metagenomic data," *Genome Res.*, vol. 17, no. 3, pp. 377–386, Mar. 2007.

[10] A. Brady and S. L. Salzberg, "Phymm and PhymmBL: Metagenomic phylogenetic classification with interpolated Markov models," *Nature Methods*, vol. 6, no. 9, pp. 673–U68, Sep. 2009.

[11] N. J. MacDonald, D. H. Parks, and R. G. Beiko, "Rapid identification of high-confidence taxonomic assignments for metagenomic data," *Nucleic Acids Res.*, vol. 40, no. 14, Aug. 2012, Art. no. e111.

[12] J. A. Eisen, "Environmental shotgun sequencing: Its potential and challenges for studying the hidden world of microbes," *Plos Biol.*, vol. 5, no. 3, pp. 384–388, Mar. 2007.

[13] Y. W. Wu and Y. Ye, "A novel abundance-based algorithm for binning metagenomic sequences using l-tuples," *J. Comput. Biol.*, vol. 18, no. 3, pp. 523–534, Mar. 2011.

[14] H. C. Leung, et al., "A robust and accurate binning algorithm for metagenomic sequences with arbitrary species abundance ratio," *Bioinf.*, vol. 27, no. 11, pp. 1489–1495, 2011.

[15] K. C. Wrighton, "Fermentation, hydrogen, and sulfur metabolism in multiple uncultivated bacterial phyla," *Science*, vol. 338, no. 6108, pp. 742–742, 2012.

[16] J. Alneberg, B. S. Bjarnason, I. D. Bruijn, M. Schirmer, J. Quick, U. Z. Ijaz, N. J. Loman, A. F. Andersson, and C. Quince, "CONCOCT: Clustering cONtigs on COverage and ComposiTion," *Quantitative Biol.*, 2013.

[17] H. B. Nielsen, et al., "Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes," *Nature Biotechnol.*, vol. 32, no. 8, pp. 822–828, 2014.

[18] P. L. Lin, P. W. Huang, C. H. Kuo, and Y. H. Lai, "A size-insensitive integrity-based fuzzy c-means method for data clustering," *Pattern Recognit.*, vol. 47, no. 5, pp. 2042–2056, May 2014.

[19] Y. Liu, T. Hou, and F. Liu, "Improving fuzzy c-means method for unbalanced dataset," *Electron. Lett.*, vol. 51, no. 23, pp. 1880–1881, 2015.

[20] J. C. Noordam, W. H. A. M. van den Broek, and L. M. C. Buydens, "Multivariate image segmentation with cluster size insensitive fuzzy C-means," *Chemometr. Intell. Lab. Syst.*, vol. 64, no. 1, pp. 65–78, 2002.

[21] R. Tibshirani and G. Walther, "Cluster validation by prediction strength," *J. Comput. Graph. Stat.*, vol. 14, no. 3, pp. 511–528, 2005.

[22] Y. Fukuyama and M. Sugeno, "A new method of choosing the number of clusters for the fuzzy c-means method," in *Proc. 5th Fuzzy Syst. Symp.*, 1989, pp. 247–250.

[23] Y. Wang, H. C. Leung, S. M. Yiu, and F. Y. Chin, "MetaCluster 4.0: A novel binning algorithm for NGS reads and huge number of species," *J. Comput. Biol.*, vol. 19, no. 2, pp. 241–249, Feb. 2012.

[24] M. Strous, B. Kraft, R. Bisdorf, and H. E. Tegetmeyer, "The binning of metagenomic contigs for microbial physiology of mixed cultures," *Frontier Microbiol.*, vol. 3, 2012, Art. no. 410.

[25] F. Zhou, V. Olman, and Y. Xu, "Barcodes for genomes and applications," *BMC Bioinf.*, vol. 9, 2008, Art. no. 546.

[26] W. L. Cai, S. C. Chen, and D. Q. Zhang, "Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation," *Pattern Recognit.*, vol. 40, no. 3, pp. 825–838, Mar. 2007.

[27] F. d. A. T. de Carvalho and C. P. Tenório, "Fuzzy K-means clustering algorithms for interval-valued data based on adaptive quadratic distances," *Fuzzy Sets Syst.*, vol. 161, no. 23, pp. 2978–2999, 2010.

[28] J. C. Wooley, A. Godzik, and I. Friedberg, "A primer on metagenomics," *Comput. Biol.*, vol. 6, no. 6, pp. 1–13, 2010.

[29] L. M. Rodriguez-R and K. T. Konstantinidis, "Nonpareil: A redundancy-based approach to assess the level of coverage in metagenomic datasets," *Bioinf.*, vol. 30, no. 5, pp. 629–635, Mar. 2014.

[30] J. C. Bezdek, "Cluster validity with fuzzy sets," *Int. J. Cybern. Syst.*, vol. 3, no. 3, pp. 58–73, 1973.

[31] K. L. Wu and M. S. Yang, "A cluster validity index for fuzzy clustering," *Pattern Recognit. Lett.*, vol. 26, no. 9, pp. 1275–1291, Jul. 2005.

[32] R. Mondav, et al., "Discovery of a novel methanogen prevalent in thawing permafrost," *Nature Commun.*, vol. 5, p. 3212, Feb. 2014.

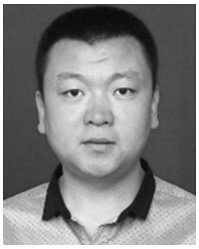
[33] D. R. Zerbino and E. Birney, "Velvet: Algorithms for de novo short read assembly using de Bruijn graphs," *Genome Res.*, vol. 18, no. 5, pp. 821–829, May 2008.

[34] C. Camacho, et al., "BLAST+: Architecture and applications," *BMC Bioinf.*, vol. 10, no. 1, p. 421, 2009.

[35] R. Li, et al., "De novo assembly of human genomes with massively parallel short read sequencing," *Genome Res.*, vol. 20, no. 2, pp. 265–272, 2010.

Yun Liu received the BS degree in 2011 from the Department of Control Science and Engineering, Jilin University, Changchun, China. He is currently working toward the PhD degree at the same affiliation. His areas of research include pattern recognition, data clustering, and metagenomic binning method.

Tao Hou received the BS degree from the College of Mathematics, and MS and PhD degrees from the College of Communication and Engineering, Jilin University. Now, she is a lecturer at Jilin University. Her areas of research include machine learning and bioinformatics.



Bing Kang received the BS degree from the Changchun University of Technology and MS degree from Jilin University. Now, he received the PhD degree from the Department of Control Science and Engineering, Jilin University. His areas of research include pattern recognition and bioinformatics.

Fu Liu received the BS and MS degrees from the Jilin University of Technology in 1991 and 1994, respectively, and the PhD degree from the Department of Control Science and Engineering, Jilin University, in 2002. He is currently a professor at Jilin University. His research interests include machine vision, pattern recognition, bioinformatics, and biometrics.