Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Genome analysis

Supplementary file of 'BMC3C: Binning Metagenomic Contigs using Codon usage, sequence Composition and read Coverage'

Guoxian Yu [1]*+, Yuan Jiang[1]+, Jun Wang [1] , Hao Zhang [2] and Haiwei Luo [2]*

[1]College of Computer and Information Science, Southwest University, Chongqing, China.
[2]School of Life Sciences and Partner State Key Laboratory of Agrobiotechnology, The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China;

*gxyu@swu.edu.cn(G-X. Yu);hluo2006@gmail.com (H. Luo)

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Metagenomics investigates the DNA sequences directly recovered from environmental samples. It often starts with reads assembly, which leads to contigs rather than more complete genomes. Therefore, contig binning methods are subsequently used to bin contigs into operational taxonomic units (OTUs). While some clustering-based binning methods have been developed, they generally suffer from problems related to stability and robustness.

Results: We introduce BMC3C, an ensemble clustering-based method, to accurately and robustly bin contigs by making use of DNA sequence Composition, Coverage across multiple samples, and Codon usage. BMC3C begins by searching the proper number of clusters and repeatedly applying the k-means clustering with different initializations to cluster contigs. Next, a weight graph with each node representing a contig is derived from these clusters. If two contigs are frequently grouped into the same cluster, the weight between them is high, and otherwise low. BMC3C finally employs a graph partitioning technique to partition the weight graph into subgraphs, each corresponding to a genome bin. We conduct experiments on both simulated and real-world datasets to evaluate BMC3C, and compare it with the state-of-the-art binning tools. We show that BMC3C has an improved performance than these tools. To our knowledge, this is the first time that the codon usage features and ensemble clustering are used in metagenomic contig binning and lead to improved performance of binning methods.

Availability: The codes of BMC3C are available at http://mlda.swu.edu.cn/codes.php?name=BMC3C.

Contact: gxyu@swu.edu.cn(G-X. Yu);hluo2006@gmail.com (H. Luo)

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Dataset

The **Sharon** dataset includes 18 libraries sequenced from 11 fecal microbiome samples from a premature infant, 7 of which were subject to re-sequencing. This dataset was downloaded from NCBI (SRA052203). The **HMP** dataset contains 10 samples. They were downloaded from the HMP website (http://hmpdacc.org/), and the SRS IDs of the ten reads files are SRS014689, SRS014691, SRS015378, SRS016203, SRS018778, SRS019027, SRS055426, SRS062540, SRS063215, and SRS064645. The **Human gut** dataset includes 264 human gut microbial samples, which were downloaded from NCBI (ERP000108). The **Amazon** dataset contains three Amazon River plume samples(SRR1182512, SRR1186214, and SRR1199271). The **COPD** dataset contains eight sputum samples from eight COPD patients (ERS799128, ERS799129, ERS799130, ERS799131, ERS799132, ERS799133, ERS799134, and ERS799135). The later two datasets were downloaded from NCBI.

2 Evaluation Criteria

We use three representative evaluation metrics, *precision*, *recall*, and Adjusted Rand Index (*ARI*), to evaluate the clustering results. These metrics are calculated from True Positive (*TP*), False Positive (*FP*), False Negative (*FN*), and True Negative (*TN*). *TP* and *FP* are the number of pairwise contigs with the same labels being clustered into the same and different clusters, respectively. *FN* and *TN* are the number of pairwise contigs with different labels being clustered into the same and different clusters, respectively. Next, we define a matrix $\mathbf{Z} \in \mathbb{R}^{k^* \times c}$, where k^* is the number of clusters obtained by a particular clustering method, c is the number of ground-truth cluster labels, and $\mathbf{Z}_{h,j}$ represents the shared number of contigs between the h-th cluster and the j-th label. The Rand index (*RI*) is a measure of the similarity between real labels and results of clustering. The Adjusted Rand Index (*ARI*) is the corrected-for-chance version of the *RI*, and it may yield negative values for low-quality clustering. *ARI* is defined as follow:

$$ARI = \frac{2(TP \times TN - FP \times FN)}{FP^2 + FN^2 + 2TP \times TN + (TP + TN) \times (FP + FN)} \quad (1)$$

$Precision$ and $recall$ are calculated as:

$$precision = \frac{1}{n} \sum_{h=1}^{k^*} max_j \left\{ \mathbf{Z}_{h,j} \right\} \quad (2)$$

$$recall = \frac{1}{n} \sum_{j=1}^{c} max_h \left\{ \mathbf{Z}_{h,j} \right\} \quad (3)$$

3 The effect of ensemble k-means

BMC3C utilizes multiple clusterings obtained by repeating k-means to build a graph. To evaluate BMC3C under different numbers of base clusterings, we increase the number of base clusterings from 5 to 60 with a step size of 5 (Figure S1).

We show that BMC3C has a reduced performance when the number of base clusterings is less than 30. This is largely because k-means initializes the centroids randomly, and some bad initializations lead to unreliable results and thus reduce the overall performance. Among the three evaluation metrics, ARI is less stable than $precision$ and $recall$. This is because ARI depends on TP, FP, FN and TN, and it decreases when either FP or FN increases (see Eq. (1)). Thus, *ARI* has a greater discriminative power than *precision* and *recall*. As the number of base clusterings exceeds 45, BMC3C tends to be stable, but a further increase of this number reduces the computing efficiency. We thus choose 50 as the number of base clusterings in the present study.

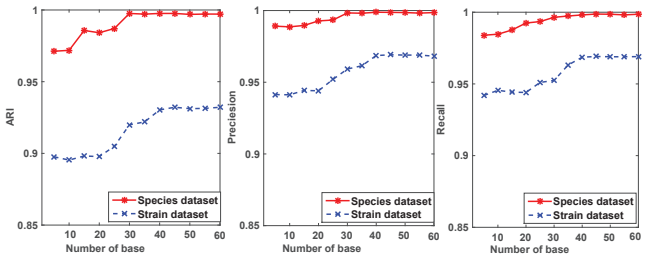

Fig. S1. Results of BMC3C with different numbers of base clusterings (k-means). The horizontal axis is the number of base clusterings for ensemble clustering.

To demonstrate the improved performance of our ensemble method compared to the base k-means method, we repeat k-means 50 times and compare the average and optimal results of this separate experiment with the ensemble result, which integrates 50 base k-means clusterings. For all three metrics, the ensemble method BMC3C performs significantly better than the average run of these base k-means clustering in both **Species** and **Strain** datasets (Figure S2). Even the optimal run of these base k-means clusterings is chosen as the comparison, the conclusion remains that BMC3C performs better for both **Species** (the ARI, $precision$, and $recall$ of the optimal k-means run are 0.8814, 0.8750, and 0.7240, respectively) and **Strain** (the best ARI, $precision$, and $recall$ of k-means are 0.9037, 0.9339, and 0.9361, respectively) datasets. This comparison shows that the ensemble clustering indeed significantly improves the robustness and accuracy of single clusterings, and thus produces better performance in binning contigs.

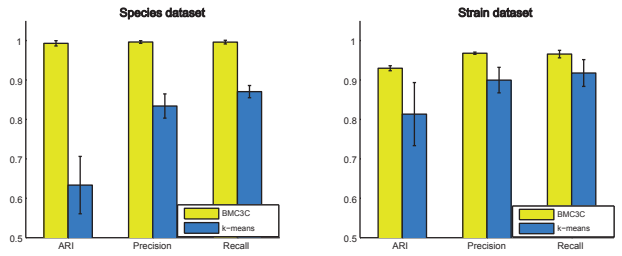

Fig. S2. Results of BMC3C and base clusterings (k-means) on two simulated datasets are shown with regards to ARI, Precision, and Recall. Left : Species dataset; Right : Strain dataset.

Next, we compare the performance of BMC3C when the integrated base clusterings use different numbers of clusters (k). We increase k in base clustering from $2k^*$ to $20k^*$ with a step size of $2k^*$, where k^* is the final number of clusters in the ensemble result. We show that the performance of BMC3C is unsatisfactory when k is smaller than $8k^*$, and reaches a plateau in all three metrics when k is greater than $10k^*$ (Figure S3). A possible reason is that a small k cannot reveal the underlying local structure of the dataset. Thus, we choose $10k^*$ as the number of clusters in base clustering, but users can set other values of k.

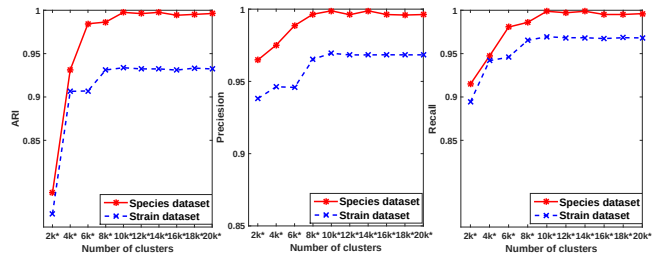

Fig. S3. Results of BMC3C with different numbers of clusters (k) in base clustering (k-means). The horizontal axis is the number of clusters in base clustering. k^* is the final number of clusters, $2k^*$ means the number of clusters (k) in base clustering is the twice of k^*.

4 Sensitivity analysis of weights

We further investigate how different weights of the three distinct features (Composition, Coverage, and Codon usage) affect the performance of BMC3C. We increase each weight from 0 to 1 with a step size of 0.2. For example, the weight 0 for the Composition features means this type of features is excluded and the other two types of features are using the same weight for concatenation, namely $\mathbf{X} = [0.5 * \widetilde{\mathbf{R}}, 0.5 * \widetilde{\mathbf{Y}}]$ are used for

binning contigs. Likewise, 0.2 means the Composition features have the weight of 0.2 and the other two each have the weight 0.4 to concatenate the feature vector $\mathbf{X} = [0.2 * \widetilde{\mathbf{H}}, 0.4 * \widetilde{\mathbf{R}}, 0.4 * \widetilde{\mathbf{Y}}]$. As shown in Figure S4, BMC3C generally has the best performance when all three types of features are used. However, different weights affect its performance. For the **Sharon** and the **COPD** datasets, BMC3C has stable performance with varying weights of features. Particularly, BMC3C has reduced performance when one or two types of features are used. As to the **Amazon** dataset, BMC3C has the best performance when the weight of the Composition features is set as 0, and its performance is decreased with the weight of the Composition features increasing. The possible reason is that some bacteria are too similar in sequence and have similar Composition features. On the other hand, BMC3C has increased performance as the weight of the Composition features increases in the **HMP** dataset, suggesting that the Composition features are more helpful in this dataset than the other two types of features. For the **Human gut** dataset, BMC3C also has stable performance when all three types of features are used, and has reduced performance when using only one or two types of features. In summary, the Coverage and Composition features are more informative than the Codon usage features. However, Codon usage is a highly desirable feature, since it further increases the performance of BMC3C. We observe that tuning the weights for these types of features improves the performance in binning contigs. However, different types of datasets have different optimal weight assignments. For simplicity, BMC3C equally weights these types of features, but its performance can be further improved by tuning the weights.

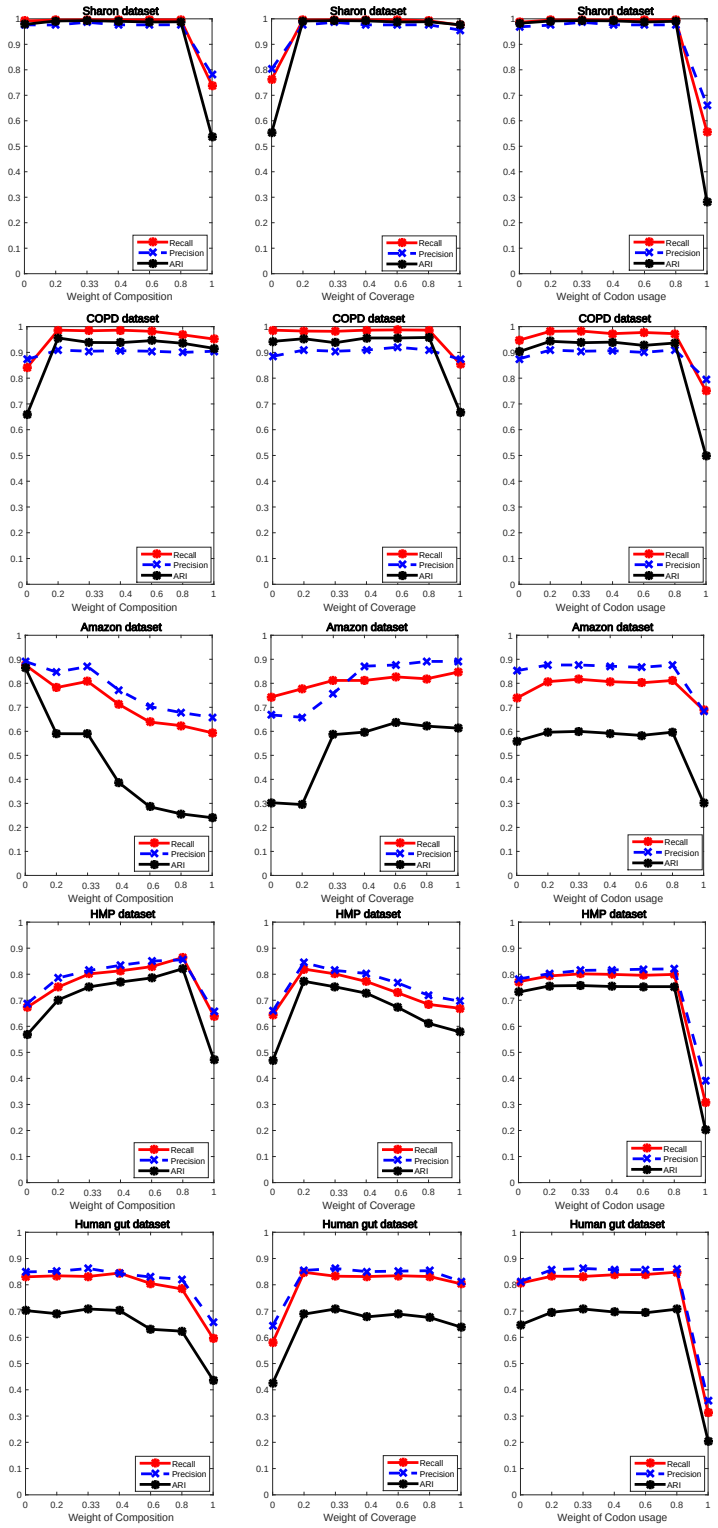

Fig. S4. Results of BMC3C under different weights of three types (Composition, Coverage and Codon usage) of features on the five real world datasets. 'Weight of Composition' represents the weight of Composition features in concatenating the numeric feature matrix $\mathbf{X}$. In 'Weight of Composition', 0 means the Composition features are excluded and $\mathbf{X}$ is constructed based on the Coverage features and the Codon usage features with equal weight as $0.5 = (1-0)/2$, namely $\mathbf{X} = [0.5 * \widetilde{\mathbf{R}}, 0.5 * \widetilde{\mathbf{Y}}]$; 1 means only the Composition features are used to construct $\mathbf{X} = [\widetilde{\mathbf{H}}]$; 0.2 means the Composition features have the weight as 0.2, and the other two types of features have the equal weight as $0.4 = (1-0.2)/2$, $\mathbf{X} = [0.2 * \widetilde{\mathbf{H}}, 0.4 * \widetilde{\mathbf{R}}, 0.4 * \widetilde{\mathbf{Y}}]$. Note, when the weight of the Composition features is close to $1/3$, $\mathbf{X}$ is constructed using the three types of features with equal weight. The 'Weight of Coverage' and 'Weight of Codon usage' follow the similar notations and meanings.