Improving contig binning of metagenomic data using $d^S_2$ oligonucleotide frequency dissimilarity

Ying Wang1*, Kun Wang1, Yang Young Lu2 and Fengzhu Sun2,3*

Abstract

Background: Metagenomics sequencing provides deep insights into microbial communities. To investigate their taxonomic structure, binning assembled contigs into discrete clusters is critical. Many binning algorithms have been developed, but their performance is not always satisfactory, especially for complex microbial communities, calling for further development.

Results: According to previous studies, relative sequence compositions are similar across different regions of the same genome, but they differ between distinct genomes. Generally, current tools have used the normalized frequency of $k$-tuples directly, but this represents an absolute, not relative, sequence composition. Therefore, we attempted to model contigs using relative $k$-tuple composition, followed by measuring dissimilarity between contigs using $d^S_2$. The $d^S_2$ was designed to measure the dissimilarity between two long sequences or Next-Generation Sequencing data with the Markov models of the background genomes. This method was effective in revealing group and gradient relationships between genomes, metagenomes and metatranscriptomes. With many binning tools available, we do not try to bin contigs from scratch. Instead, we developed $d^S_2$Bin to adjust contigs among bins based on the output of existing binning tools for a single metagenomic sample. The tool is taxonomy-free and depends only on $k$-tuples. To evaluate the performance of $d^S_2$Bin, five widely used binning tools with different strategies of sequence composition or the hybrid of sequence composition and abundance were selected to bin six synthetic and real datasets, after which $d^S_2$Bin was applied to adjust the binning results. Our experiments showed that $d^S_2$Bin consistently achieves the best performance with tuple length $k = 6$ under the independent identically distributed (i.i.d.) background model. Using the metrics of recall, precision and ARI (Adjusted Rand Index), $d^S_2$Bin improves the binning performance in 28 out of 30 testing experiments (6 datasets with 5 binning tools). The $d^S_2$Bin is available at https://github.com/kunWangkun/d2SBin.

Conclusions: Experiments showed that $d^S_2$ accurately measures the dissimilarity between contigs of metagenomic reads and that relative sequence composition is more reasonable to bin the contigs. The $d^S_2$Bin can be applied to any existing contig-binning tools for single metagenomic samples to obtain better binning results.

Keywords: Metagenomics, Contig binning, Taxonomy-independent, $d^S_2$ dissimilarity, $k$-tuple

* Correspondence: wangying@xmu.edu.cn; fsun@usc.edu
1Department of Automation, Xiamen University, Xiamen, Fujian 361005, China
2Molecular and Computational Biology Program, University of Southern California, Los Angeles, California, CA 90089, USA
Full list of author information is available at the end of the article

© The Author(s). 2017 Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated.
Background

Metagenomics sequencing provides deep insights into microbial communities [1]. A key step toward investigating their taxonomic structure within metagenomics data involves assigning assembled contigs into discrete clusters known as bins [2]. These bins represent species, genera or higher taxonomic groups [3]. Therefore, efficient and accurate binning of contigs is essential for metagenomics studies.

The binning of contigs remains challenging owing to repetitive sequence regions within or across genomes, sequencing errors, and strain-level variation within the same species [4]. Many studies have reported on binning, essentially highlighting two different strategies [5]: “taxonomy-dependent” supervised classification and “taxonomy-independent” unsupervised clustering. “Taxonomy-dependent” studies are based on sequence alignments [6], phylogenetic models [7, 8] or oligonucleotide patterns [9]. “Taxonomy-independent” studies extract features from contigs to infer bins based on sequence composition [10–14], abundance [15], or hybrids of both sequence composition and abundance [4, 5, 16–18]. Therefore, these approaches can be applied to bin contigs from incomplete or uncultivated genomes. Some hybrid binning tools, such as COCACOLA [5], CONCOCT [4], MaxBin2.0 [18] and GroopM [16], are designed to bin contigs based on multiple related metagenomic samples. Contigs with similar coverage profiles are more likely to come from the same genome. Previous studies showed that co-variant coverage profiles across multiple related metagenomes play important roles in contig binning [4, 5]. The multiple related samples should be temporal or spatial samples of a given ecosystem [16] composed of similar microbial organisms, but different abundance levels. However, in many situations, multiple related samples may not be available in the required numbers, and as a result, contig-binning based on single metagenomes is still important.

Contig binning tools based on a single sample generally follow one of three strategies. 1) Sequence composition. It is usually denoted as frequencies of k-tuples (k-mers) with k= 2–6 as genomic signatures of contigs. MetaWatt [12] and SCIMM [11] built multivariate statistics and/or interpolated Markov models of background genomes to bin the contigs. Metacluster 3.0 [14] clustered the contigs using k-tuple frequency and Spearman correlation between the k-tuple frequency vectors. LikelyBin [10] utilized Markov Chain Monte Carlo approaches based on 2- to 5-tuples. 2) Abundance. AbundanceBin [15] estimated the relative abundance levels of species living in the same environment based on Poisson distributions of 20-tuples with an Expectation Maximization (EM) algorithm. The MBBC [19] package estimated the abundance of each genome using the Poisson process. All tools based on abundance are designed to bin short or long reads instead of assembled contigs. 3) Hybrid of composition and abundance. Maxbin1.0 [17] combined 4-tuple frequencies and scaffold coverage levels to populate the genomic bins using single-copy marker genes and an Expectation Maximization (EM) algorithm. MyCC [20] combined genomic signatures, marker genes and optional contig coverages within one or multiple samples.

Contig binning using k-tuple composition is based on the observation that relative sequence compositions are similar across different regions of the same genome, but differ between distinct genomes [21, 22]. The frequency vector of k-tuples is one of the representation of sequence composition. In general, current tools use the frequency of k-tuples directly, but this represents absolute, not relative, sequence composition. Here, “absolute” frequency refers to the number of occurrences of a k-tuple over the total number of occurrences of all k-tuples. On the other hand, “relative” frequency refers to the difference between the observed frequency of a k-tuple and the corresponding expected frequency under a given background model. Contigs in the same bin are from the same taxonomic group, such as one class, species or strain. Therefore, contigs from the same bin are expected to obey a consistent background model. Several sequence dissimilarity measures based on relative frequencies of k-tuples have been developed such as CVTree, $d^*_2$ and $d^s_2$, and recent studies [23–27] have shown that $d^s_2$ is superior to other dissimilarity measures for the comparison of genome sequences based on relative k-tuple frequencies. Therefore, in the present study, we attempted to model the relative sequence composition and measure dissimilarity between contigs with $d^s_2$ for a single metagenomic sample. The $d^s_2$ was designed to measure the dissimilarity between two sequences or next generation sequencing data by modeling the background genomes [23] using Markov and interpolated Markov chains. Previous studies verified the effectiveness of $d^s_2$ in revealing group and gradient relationships between genomes [24, 25], metagenomes [28] and metatranscriptomes [26, 27]. However, binning of contigs directly using $d^s_2$ is computationally expensive and impractical for large metagenomics studies due to the need to construct Markov background models for sequences and to calculate the expected counts of k-tuples. On the other hand, many binning tools based on absolute k-tuple frequencies and the results from such methods are reasonable. Still, these tools and methods can be improved by using $d^s_2$ dissimilarity. Therefore, in the present study, we do not bin the contigs from scratch. Instead, we attempt to adjust contig bins based on the output of any existing binning tools. We model each contig with a Markov chain based on its k-tuple frequency vector. The bin’s center is
sequences based on either long genomic sequences or NGS short reads in which expected word counts are subtracted from the observed counts for each sequence. The background adjusted word counts are then compared using correlation to measure the dissimilarity between the two sequences [25]. Let \( c_X = (c_{X,1}, c_{X,2}, \ldots, c_{X,d^s}) \) and \( c_Y = (c_{Y,1}, c_{Y,2}, \ldots, c_{Y,d^s}) \) be the \( d^s \)-tuple frequency vectors from two sequences \( X \) and \( Y \), respectively, where \( c_{X,i} \) is the occurring times of the \( i \)-th \( d^s \)-tuple in sequence \( X \) and \( i = 1 \ldots 4^s \). At each base in the tuple, there are four possible nucleotides, that is A, C, G, and T, for nucleotide sequences. So there are \( 4^k \) combinations when tuple length is \( k \).

The \( d^s \)-diagram is defined as

\[
d^s_2(\hat{c}_X, \hat{c}_Y) = \frac{1}{2} \left( 1 - \frac{D^s_2(\hat{c}_X, \hat{c}_Y)}{\sqrt{\sum_{i=1}^{4^s} \frac{c_{X,i}^2}{c_{X,i}^2 + c_{Y,i}^2}} \sqrt{\sum_{i=1}^{4^s} \frac{c_{Y,i}^2}{c_{X,i}^2 + c_{Y,i}^2}}} \right),
\]

where

\[
D^s_2(\hat{c}_X, \hat{c}_Y) = \sum_{i=1}^{4^s} \frac{\hat{c}_{X,i} \hat{c}_{Y,i}}{\sqrt{\hat{c}_{X,i}^2 + \hat{c}_{Y,i}^2}},
\]

\[
\hat{c}_{X,i} = c_{X,i} - n_X p_{X,i}, \quad \hat{c}_{Y,i} = c_{Y,i} - n_Y p_{Y,i},
\]

where \( p_{*,i} \) is the probability of the \( i \)-th \( d^s \)-tuple under the Markov model with order \( r = 0 \) - 3 for one long sequence or set of reads and \( n_* = \sum_{i=1}^{4^s} c_{*,i} \), * is \( X \) or \( Y \) is the sum of occurrences of all \( d^s \)-tuples. The value of \( d^s \) is between 0 and 1. The \( p_{X,i} \) is the probability of the \( i \)-th \( d^s \)-tuple under the background sequence for \( X \). The \( p_{X,i} \) can be the probability under the i.i.d. model, or under the Markov chain of different orders. The \( d^s \)-tuple is denoted as \( w = w_1w_2 \ldots w_k \). Under the \( r \)-th order Markov chain \( M_r \), the probability of the \( k \)-tuple \( w \), namely the expected frequency, can be computed as

\[
p(w|w_r) = \begin{cases} 
\prod_{j=1}^{r} p(w_j) & r = 0 \\
\prod_{j=1}^{r} p(w_{j_1}w_{j_2} \ldots w_{j_{r-1}}) & 1 \leq r \leq k - 1 
\end{cases}
\]

where \( p(w_j) \) is the probability of \( w_j \) estimated by the ratio of the number of occurrences of \( w_j \) over the number of all nucleotides. The value of \( p(w_1w_2 \ldots w_r) \) is estimated by the ratio of the number of occurrences of \( w_1w_2 \ldots w_r \) over all the number of \( r \)-tuple occurrences. The value of \( p(w_{j_1}w_{j_2} \ldots w_{j_{r-1}}) \) is estimated by the fraction of occurrences of \( w_{j_1}w_{j_2} \ldots w_{j_{r-1}} \) conditional on the previous occurrences of \( w_{j_1}w_{j_2} \ldots w_{j_{r-1}} \).
$d_{2}^{S}$Bin: Contig binning based on the $d_{2}^{S}$ measure

Let $S = \{S_1, S_2, \cdots, S_l\}$ be the partition of all contigs into $l$ bins. Contig $X$ is represented as $c_X = (c_{X,1}, c_{X,2}, \cdots, c_{X,k})$, the occurrence vector of $k$-tuples within the contig. The center of bin $S_j$ is represented as the average frequency vector,

$$c_{S_j} = \frac{1}{n_j} \sum_{X \in S_j} c_X,$$

where $X_i$ is the contig currently in $S_j$ and $n_j$ is the number of contigs in $S_j$. The value of $d_{2}^{S}(c_X, c_{S_j})$ quantifies the dissimilarity between contig $X$ and bin $S_j$.

In our study, when the number of bins is fixed, the metrics of binning call for minimizing the within-bin sum of squares based on $d_{2}^{S}$ dissimilarity, that is,

$$\arg \min_{S} \sum_{j=1}^{l} \sum_{X \in S_j} d_{2}^{S}(\hat{c}_X, \hat{c}_{S_j}).$$
We then used the K-means clustering algorithm to optimize Eq. (6).

**Experimental design**

The purpose of our study is to improve binning results using $d^2_{S2}$Bin based on the output of current existing binning tools. Therefore, we adopted both synthetic and real testing datasets generated, or used, by previous binning tools in order to test the performance of $d^2_{S2}$Bin, as shown in Table 1. The $d^2_{S2}$Bin was applied to the binning results of five contig-binning tools, respectively, to evaluate its performance in improving their binning results.

**Selection of contig binning tools**

The $d^2_{S2}$Bin was applied to adjust the contig-binning results from MaxBin1.0 [17], MetaCluster3.0 [14], MetaWatt [12], MyCC [20] and SCIMM [11] to evaluate its performance. These five widely used contig-binning tools use different binning strategies to bin the contigs for single metagenomic sample: 1) Sequence composition: MetaCluster3.0 [14] measures the Spearman distance between 4-tuple frequency vectors and bins contigs with the K-median algorithm. The MetaCluster4.0 [29] and 5.0 [30] were designed to bin the reads from metagenomics samples of different abundance characteristics. MetaWatt [12] and SCIMM [11] build interpolated Markov models of the background genomes and assign the contigs to bins with maximum likelihood. 2) Hybrid of abundance and sequence composition: MaxBin1.0 [17] measures the Euclidean distance between 4-tuple frequency vectors of contigs and assigns them with an EM algorithm, taking scaffold coverage levels into consideration. MyCC [20] combines genomic signatures, marker genes and optional contig coverages within one or multiple samples.

**Five synthetic testing datasets with 10 genomes and 100 genomes**

MaxBin1.0 [17] used these five datasets to evaluate its performance. Here we used the same five datasets to evaluate the performance of $d^2_{S2}$Bin. Short reads were simulated by MetaSim [31] and assembled to contigs by Velvet [32]. The contigs and their labels are available for downloading from the MaxBin1.0 paper [17]. For the metagenomes containing 10 genomes, 5 million and 20 million paired-end reads were sampled as 20x and 80x average coverage, respectively. For the metagenomes containing 100 genomes, 100 million paired-end reads were sampled with three settings to create simLC+, simMC+ and simHC+. The three datasets represent microbial communities with different levels of complexity, which mimicked the setting of the previous study [33]; simLC simulates low-complexity communities dominated by a single near-clonal population flanked by low-abundance ones. Such datasets result in a near-complete draft assembly of the dominant population in, for example, bioreactor communities [34]. simMC resembles moderately complex communities with more than one dominant population, also flanked by low-abundance ones, as has been observed in an acid mine drainage biofilm [35] and Olavius algarvensis symbionts [36]. These types of communities usually result in substantial assembly of the dominant populations according to their clonality. simHC simulates high-complexity communities lacking dominant populations, such as agricultural soil [37], where no dominant strains are present and minimal assembly results. In addition, the empirical 80-bps error model, which incorporates different error types (deletion, insertion, substitution) at certain positions with empirical error probabilities for Illumina, was produced by MetaSim [31] and used in simulating all metagenomes [17].

**One real testing dataset, Sharon**

This dataset was applied to test the binning tools COCACOLA [5] and CONCOCT [4]. The dataset is composed of a time-series of 11 fecal microbiome samples from a premature infant [38], denoted as ‘Sharon’. All metagenomic sequencing reads from the 11 samples were merged together, and 5579 contigs were assembled. The contigs were annotated with TAXAassign [39], and 2614 contigs were unambiguously aligned to 21 species [5].

The above datasets cover various species diversity, species dissimilarity, sequencing depth, and community

---

**Table 1 Synthetic and real testing datasets for contig binning**

| Testing datasets | Tools tested previously | Tools tested in this study |
|------------------|-------------------------|---------------------------|
| Synthetic        |                         |                           |
| 10 genomes: 20x  | Maxbin 1.0 [17]         | Maxbin 1.0 [17] +$d^2_{S2}$Bin |
| 10 genomes: 80x  |                         |                           |
| 100 genomes: simLC+ | MetaCluster 3.0 [14]    |                           |
| 100 genomes: simMC+ | MetaWatt3.5.3 [3]       |                           |
| 100 genomes: simHC+ | SCIMM 0.3.0 [11]       |                           |
| Real             |                         |                           |
| Sharon           | COCACOLA [5]            | MyCC_2017 [20]            |
|                  | CONCOCT [4]             |                           |
complexity. They include synthetic and real data. Therefore, testing on these datasets would yield a comprehensive evaluation of $d^S_2$Bin.

**Evaluation criteria**

To evaluate the performance of $d^S_2$Bin, three commonly used criteria in binning studies [4, 5, 17], recall, precision and ARI (Adjusted Rand Index), were applied in our study. As described in COCACOLA [5], the binning result is represented as a $K \times S$ matrix $A = (a_{ks})$ with $K$ bins on $S$ species where $a_{ks}$ indicates the shared number of contigs between the $k^{\text{th}}$ bin and the $s^{\text{th}}$ species. Each contig binning tool filters out low-quality contigs; therefore, $N$ is the total number of contigs passing through the filter and binned by the tools.

**Recall:** For each species, we first find the bin that contains the maximum number of contigs from the species. We then sum over the maximum number of all species and divide by the number of contigs.

$$\text{recall} = \frac{1}{N} \sum_s \max_k \{a_{ks}\}$$  \hspace{1cm} (7)

**Precision:** For each contig bin, we first find the species with the maximum number of contigs assigned to the bin. We then sum the maximum numbers across all bins and divide by the number of contigs.

$$\text{precision} = \frac{1}{N} \sum_k \max_s \{a_{ks}\}$$  \hspace{1cm} (8)

ARI (Adjusted Rand Index): ARI is a unified measure of clustering results to determine how far from that perfect grouping a bin result falls. ARI focuses on whether pairs of contigs belonging to the same species can be binned together or not. The detailed descriptions can be found in [4, 5].

$$\text{ARI} = \frac{\sum_ks \frac{a_{ks}}{2} t_3}{\frac{1}{2} (t_1 + t_2) t_3}$$  \hspace{1cm} (9)

where $t_1 = \sum_s \left( \frac{a_s}{2} \right)$, $t_2 = \sum_k \left( \frac{a_k}{2} \right)$, $t_3 = \frac{2rt_s}{N}$ and $a_{ks} = \sum_s a_{ks}$, $a_k = \sum_k a_{ks}$.

**Results**

In the calculation of $d^S_2$ dissimilarity, the setting of tuple length for $k$-tuple and Markov order for the background sequences are required. Based on previous studies [4, 5], for $d^S_2$, tuple length $k$ was generally set to 4–7 tuples, and the order of Markov chain was generally set as 0–2, as in previous applications, to analyze metagenomic and metatranscriptomic samples [25, 26]. Therefore, we extended the testing range of tuple length and Markov order as 4–8 and 0–3 to assess the effect of tuple length and Markov order for $d^S_2$Bin on contig binning. As shown in Table 2, for the binning results of MaxBin on 10genome-80x, the i.i.d. (that is 0-order Markov) model obtained the highest three indexes at almost all tuple lengths. The models based on tuple length $k = 6$ represent superior performance. The best performance was achieved under the i.i.d. background model of 6-tuples. All three criteria dropped suddenly at $k = 8$. The experiment offered initial guidance for the selection of tuple length and Markov order.

**Length selection of $k$-tuple in $d^S_2$Bin**

According to Table 2, we calculated $d^S_2$ with 4-8 bp tuples under the i.i.d. model based on the output of the existing binning tools. These tools were run under their default tuple length and mode. The datasets 10genome 80x and 100genome-simHC+ were selected to test the effect of tuple length on the performance of $d^S_2$Bin. For both datasets, $d^S_2$Bin based on 6-tuples achieved the best performance on precision, recall and ARI for all five

| 10genome-80x | Recall(%) | Precision(%) | ARI(%) |
|--------------|-----------|--------------|--------|
| MaxBin       | 93.48     | 93.48        | 90.96  |
| MaxBin+$d^S_2$Bin | 96.42     | 96.42        | 95.57  |
|    k = 4     | r = 0     | 96.42        | 95.57  |
|    r = 1     | 93.99     | 93.99        | 90.86  |
|    r = 2     | 86.35     | 86.35        | 76.18  |
|    r = 5     | 96.83     | 96.83        | 96.03  |
|    r = 6     | 96.93     | 96.93        | 96.05  |
|    r = 7     | 96.93     | 96.93        | 96.05  |
|    r = 8     | 96.01     | 96.01        | 94.57  |
|    r = 2     | 94.24     | 94.24        | 91.40  |
|    r = 3     | 81.28     | 83.77        | 71.56  |
|    r = 7     | 94.41     | 94.41        | 92.08  |
|    r = 8     | 88.28     | 88.28        | 90.04  |
|    r = 1     | 87.17     | 87.17        | 84.78  |
|    r = 2     | 87.19     | 87.12        | 82.73  |
|    r = 3     | 60.08     | 73.08        | 46.46  |

The optimal numbers with respect Markov order are in bold.
tools. Figures 2 and 3 only plot the curves of tuple length \(k = 4–6\) because the severe dropping in performance with \(k = 7, 8\) led to an excessively wide Y-axis coordinate range, and the curves of \(k = 4–6\) appeared to aggregate, making it hard to display the superiority of \(k = 6\). Therefore, we set \(k = 6\) with \(d^2_{\text{Bin}}\) in the rest of our study.

**Order selection for Markov chain in \(d^2_{\text{Bin}}\)**

To obtain the most suitable Markov order for the background genome, we fixed the tuple length \(k = 6\) and applied 0-2nd order Markov chain to calculate \(d^2_{\text{Bin}}\) for datasets 10genome 80× and 100genome-simHC+ on the output of five contig-binning tools. As shown in Figs. 4 and 5, for both datasets, \(d^2_{\text{Bin}}\) under the i.i.d. model of 6-tuple achieves the best performance for \(\text{Precision}, \text{Recall}\) and \(\text{ARI}\) on all five tools. According to our previous studies about applying \(d^2_{\text{Bin}}\) to compare metagenomic [28] and metatranscriptomic samples [26], \(d^2_{\text{Bin}}\) under the i.i.d. model always achieved best results for all the 12 testing datasets, which illustrated that the i.i.d. model works well for the study of microbial communities. This is probably due to the fact that each bin is a mixture of several genomes and no Markov chain models with fixed order greater than 0 can describe the bin better. Therefore, we set tuple length \(k = 6\) and the i.i.d. model in \(d^2_{\text{Bin}}\).

**Experiments on contig binning**

The contig-binning tools Maxbin [17], Metacluster 3.0 [14], MetaWatt [3], SCIMM [11] and MyCC [20] were applied to bin the contigs from the six synthetic and real datasets with their original running modes. Based on the results from these tools, \(d^2_{\text{Bin}}\) was further applied to adjust the contigs among bins. \(d^2_{\text{Bin}}\) did not change the number of bins obtained by the original tools. The bar graphs in Fig. 6 illustrate the \(\text{Recall}, \text{Precision}\) and \(\text{ARI}\) of the output of the five existing tools and after the adjustment of \(d^2_{\text{Bin}}\) for the six datasets. In most cases, the three criteria were improved by 1%–22%. Additional file 1: Table S1 presents the numerical values of the three indexes and offers more detailed information on all experiments, including the number of total\&binned contigs and actual\&clustered bins, providing more comprehensive view about the scale of dataset, complexity and original binning performance.

**Contig binning on synthetic dataset 10 genome 80× coverage**

From Fig. 6a, it is easy to see that the three criteria were improved for all five tools. As shown in Additional file 1: Table S1, 8022 contigs were assembled from simulated metagenomic reads. The best results were obtained on MyCC where \(d^2_{\text{Bin}}\) increased \(\text{recall}, \text{precision}\) and \(\text{ARI}\).
from 97.21%, 97.21%, and 95.58% to 97.75%, 97.75% and 96.16%, respectively. MaxBin, MetaCluster and MyCC assigned the contigs into 10 bins. MetaWatt and SCIMM obtained 27 and 8 bins, respectively, but \( d_{2}^{2} \)Bin still adjusted contigs among these bins to achieve better performance.

Contig binning on synthetic dataset 10 genome 20× coverage

Compared with 20 million reads in 10 genome 80× data, 10 genome 20× data have only 5 million reads for the 10 genomes. Fig. 6b shows that \( d_{2}^{2} \)Bin improved the binning of MaxBin, MetaWatt, SCIMM and MyCC. As shown in Additional file 1: Table S1, both MaxBin and MetaCluster only produced three bins, and most contigs belonged to the three genomes with highest abundances because most contigs from the seven low-abundance genomes were discarded during preprocessing by having short length [17]. However, the \( d_{2}^{2} \)Bin only improved precision, but not recall or ARI, on MetaCluster. In order to have a deep insight on the deterioration of binning performance, we list the number of contigs from the 10 genomes in each bin, as shown in Additional file 1: Table S2–2 for MetaCluster and MetaCluster+ \( d_{2}^{2} \)Bin. Each row of the table is one genome defined by its genome ID and corresponding genome name in NCBI and each column is the clustered bin, so the element is the number of contigs from one genome inside the current bin. Among the 1217 contigs assigned by MetaCluster, there are 1209 contigs from four dominant genomes: \textit{Flavobacterium branchiophilum}, \textit{Halothiobacillus neapolitanus}, \textit{Lactobacillus casei} and \textit{Acetobacter pasteurianus} with at least 100 contigs. But MetaCluster only output three bins: the contigs from \textit{Flavobacterium branchiophilum}, \textit{Halothiobacillus neapolitanus} and \textit{Lactobacillus casei} are dominant in the three bins, and the contigs from \textit{Acetobacter pasteurianus} are scattered into the three bins. After adjustment by \( d_{2}^{2} \)Bin, the contigs from \textit{Acetobacter pasteuerianus} were merged into the same bin as \textit{Halothiobacillus neapolitanus}. \textit{Acetobacter pasteurianus} and \textit{Halothiobacillus neapolitanus} are both from the phylum Proteobacteria. Therefore, \textit{Acetobacter pasteuerianus} is phylogenetically closer to \textit{Halothiobacillus neapolitanus} than to the other two genomes. From this point of view, \( d_{2}^{2} \)Bin indeed improved the binning of MetaCluster although the performance index did not show improvement. Additional file 1: Table S2 also gives the details of contigs’ assignments in bins before and
after $d_2$Bin for the other four tools. For MyCC in Additional file 1: TableS2–5, before using $d_2$Bin, MyCC produced 5 bins and the contigs from *Halothiobacillus neapolitanus* were assigned to bin 1 and bin 4 and bin 1 included *Halothiobacillus neapolitanus* and *Lactobacillus casei*, which lead to the low ARI index as 24.76%. After using $d_2$Bin, most contigs from *Halothiobacillus neapolitanus* were assigned to bin 4, and bin 1 mainly included contigs from *Lactobacillus casei*. The ARI was increased to 70.48%. The result demonstrates that $d_2$Bin tends to assign contigs with consistent or similar background models to the same bin.

**Contig binning on synthetic dataset 100 genome-simHC+**

SimHC+ has evenly distributed species abundance levels with no dominant species. According to Fig. 6c, the three criteria were all improved for the five tools. According to Additional file 1: Table S1, among a total of 407,873 contigs, 13,919 were clustered into 87 bins by MaxBin with 80.23%, 76.69 and 64.58% recall, precision and ARI, respectively. After $d_2$Bin, the three indexes were improved to 90.67%, 80.14% and 74.03%, respectively, showing overall superior performance. MetaCluster, MetaWatt, and MyCC produced 97, 129 and 94 bins, respectively, and recall, precision and ARI were improved for all of them by $d_2$Bin. SCIMM only clustered 19 bins, which led to low precision and ARI, but $d_2$Bin still improved the three metrics.

**Contig binning on synthetic dataset 100 genome-simMC+**

According to Fig. 6d, the three criteria were improved by $d_2$Bin for MaxBin, MetaCluster, SCIMM and MyCC. Owing to the poor assembly quality of simMC+ [17], only ~10,000+ contigs of the 795,573 passed the minimum length threshold, among which a small portion came from low-abundance genomes. Therefore, only high-abundance genomes were binned, and 11 bins were generated for MaxBin and MetaCluster, and 15 bins for MyCC. The large disparity between the number of real species and bins led to low precision and ARI. However, $d_2$Bin still greatly improved recall, precision and ARI. The exception was MetaWatt. Among the 11,987 clustered contigs, MetaWatt isolated 41 bins. In this case, extracting contigs from the dominant genome from each bin would leave only 7978, meaning that one-third of the contigs would remain to interfere with the modeling
of the 41 dominant genomes, in turn leading to decreased performance for precision and ARI.

Contig binning on synthetic dataset 100 genome-simLC+

\(d^2_{sB}Bin\) improved the binning performance for all tools. All three metrics were also significantly improved by \(d^2_{sB}\) Bin. For SCIMM, \(d^2_{sB}\)Bin increased recall, precision and ARI from 70.99%, 46.29% and 32.64% to 76.42%, 65.46% and 55.24%, respectively, which represents the best performance among the five tools.

Contig binning on real dataset Sharon

For this real dataset, the ground truth of binning was not available. The following two evaluations were implemented: (1) We only binned the 2614 contigs with unambiguous labels belonging to 21 species, and the annotations were considered as the ground truth. MaxBin, MetaCluster, MetaWatt, SCIMM and MyCC isolated 11, 10, 23, 19 and 16 bins for Sharon originally. As shown in Fig. 6f, based on their binning outputs, \(d^2_{sB}\)Bin adjusted the contig binning and increased Recall, Precision and ARI for all tools. (2) We applied CheckM [40] to estimate the approximate contamination and genome completeness of the contigs in the bins free from ground truth. Figure 7a shows the number of recovered genome bins by each method in different recall (completeness) threshold with precision (lack of contamination) > 80%. Although the tools identified 10–23 bins among the 21 species in the Sharon dataset, only 4–6 genome bins were recovered with precision > 80%. \(d^2_{sB}\)Bin did improve recall and precision. For MetaWatt and MyCC, \(d^2_{sB}\)Bin increased the number of bins with precision > 80%. For MetaCluster and SCIMM, \(d^2_{sB}\)Bin not only increased the number of bins with precision > 80% but also increased the number of bins with recall > 90%. The \(d^2_{sB}\)Bin also increased the recall of each bin for MaxBin and MyCC. Figure 7b shows the number of recovered genome bins at different precision thresholds with recall > 80%. For all tools, \(d^2_{sB}\)Bin increased the number of bins with recall > 80%. For MaxBin and MyCC, the number of bins with precision > 90% is also increased by \(d^2_{sB}\)Bin.

Testing on these synthetic and real datasets showed that \(d^2_{sB}\)Bin could achieve obvious improvement on the original outputs of the five testing tools.
Convergence of K-means iteration on \(d^2\text{Bin}\)

In order to evaluate the convergence of K-means iteration on \(d^2\text{Bin}\), we plotted the performance curves of the three indexes on randomly selected tools and datasets, as shown in Fig. 8. During our experiments with ten iterations, the three indexes increased significantly on the first iteration and reached steady state quickly. The “0” in the horizontal ordinate indicates the performance of the original binning tool. Therefore, in \(d^2\text{Bin}\), the iterations of contig binning with K-means will stop when no contigs is adjusted or the number of iterations reaches 5.

Software implementation and running

The code of \(d^2\text{Bin}\) was implemented with Python and Cython running under the Linux system. Cython is a superset of the Python language that additionally supports calling C functions, and the code can be compiled into a sharing library called by python directly. Tested on a server with 128G memory and Intel(R) Xeon(R) CPU E5–2620 v2 @ 2.10GHz with 6 CPU cores at 2.10 GHz, it takes 16 min to finish the adjustment of contig binning for \(d^2\text{Bin}\) on 6-tuples for 8022 contigs of 10 bins with 4000 bp length on average and the peak memory is 6.7GB. The source code of \(d^2\text{Bin}\) is available at https://github.com/kunWangkun/d2SBin.

Discussion

Our experiments demonstrate \(d^2\) can measure the similarity between contigs more accurately. However, \(d^2\) requires to build the background Markov model for each contig, which bring heavy computation burden. Therefore, in our study, instead of de novo binning from scratch, we attempt to adjust contig bins based on the
output of any existing binning tools for the single metagenomic sample. The computational issue can be overcome using this strategy. When there are multiple related samples available, the sequence composition contribute less than the co-varying coverage profiles across samples for contig binning and \( d_{2}^{2} \text{Bin} \) can not improve the contig binning for multiple metagenomic samples. The tools designed for multiple samples, like COCACOLA, GroopM, Concoct, MaxBin2.0, can achieve satisfactory results if multiple metagenomic samples are available.

Currently, \( d_{2}^{2} \text{Bin} \) does not merge, or split, the bins. In some situations that there may be large differences between the numbers of clustered bins and ground truth, merging and splitting the bins would improve the results. However, the algorithms to adjust the clustering number, such as ISODATA [41], require the inputs of the minimum threshold of between-class dissimilarity

---

Fig. 7 Evaluation of recall and precision of the Sharon dataset with CheckM. a The plot shows the number of recovered genome bins (X-axis) by each method (Y-axis) at different recall (completeness) thresholds (gray scale) with precision (lack of contamination) \( \geq 80\% \). b The plot shows the number of recovered genome bins (X-axis) by each method (Y-axis) at different precision thresholds (gray scale) with recall \( \geq 80\% \). It is clear that \( d_{2}^{2} \text{Bin} \) improved the recall and precision of each bin compared with the original tools. The number “0” shown on the border means that one or more value intervals were skipped because no genome was recovered in the intervals.

---

Fig. 8 Curves of the three indexes with the K-means iterations. The “0” in the horizontal ordinate reflects the output performance of the original binning tool, MetaCluster in (a) and SCIMM in (b). The three indexes increase significantly on the first iteration, followed by slight adjustment to reach steady values.
and the maximum threshold of within-class dissimilarity. These thresholds depend on the detailed taxonomic level which the investigators are interested in. Once these thresholds are given, we can combine the algorithms for merging and splitting bins with $d^S$Bin to further improve the binning results.

Conclusions
The ability of $d^S$Bin to achieve improved binning performance is based on the idea that contigs clustered into one bin will come from the same genome and that relative sequence compositions will be similar across different regions of the same genome, but differ between genomes [21, 22]. $d^S$ measures the dissimilarity between contig and the bin's center based on the Markov model of $k$-tuple sequence compositions.

Our experiments demonstrate that $d^S$Bin significantly improves binning performance in almost all cases, thus giving credence to the relative sequence composition model over the direct application of absolute sequence composition. We applied $d^S$Bin to five contig-binning tools with different binning strategies. Irrespective of the different strategies employed by the contig-binning tools, $d^S$Bin was able to achieve better performance for all tools tested. Finally, the optimal results for $d^S$Bin are always obtained on steady tuple length $k = 6$ under the i.i.d. model with no need to search for the optimal parameters.

Additional file

Additional file 1: Table S1. The file gives the numerical values of three criteria of contig binning on the experiments of the six testing datasets.

Table S2. Detailed binning results of the contigs before and after $d^S$Bin for dataset 10 genome-20× based on the five testing tools. (DOCX 38 kb)

Abbreviations
AR: Adjusted rand index; EM: Expectation maximization; i.i.d.: independent identically distributed

Acknowledgements
Not applicable.

Funding
This research is supported by the National Natural Science Foundation of China (61673324, 61503314), U.S. National Science Foundation grants (DMS-1518001, NIH R01GM120624, China Scholarship Council (201606315011) and Natural Science Foundation of Fujian (2016 J01316). The funding agencies had no role in study design, analysis, interpretation of results, decision to publish, or preparation of the manuscript.

Availability of data and materials
The $d^S$Bin source codes are available at https://github.com/kunWangKun/dSBin.

The five synthetic testing datasets were from: http://downloads.jbei.org/data/microbial_communities/MaxBin/MaxBin.html [42].

The real Sharon dataset was from the NCBI short-read archive (SRA052203).

Authors' contributions
YW and FS planned the project; YW developed the model and designed the experiments; KW realized the models and implemented the experiments; KW and YL analyzed the results; YW and FS wrote the main manuscript. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher’s Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
1. Department of Automation, Xiamen University, Xiamen, Fujian 361005, China. 2. Molecular and Computational Biology Program, University of Southern California, Los Angeles, California, CA 90089, USA. 3. Center for Computational Systems Biology, Fudan University, Shanghai 200433, China.

Received: 3 May 2017 Accepted: 11 September 2017

Published online: 20 September 2017

References
1. Riesenfeld CS, Schloss PD, Handelsman J. Metagenomics: genomic analysis of microbial communities. Annu Rev Genet. 2004;38:525–52.
2. Mande SS, Mohammed MB, Ghosh TS. Classification of metagenomic sequences: methods and challenges. Brief Bioinform. 2012;13(6):669–81.
3. Sedlar K, Kupkova K, Provaznik I. Bioinformatics strategies for taxonomy independent binning and visualization of sequences in shotgun metagenomics. Comput Struct Biotechnol J. 2017;15:48–55.
4. Alneberg J, et al. Binning metagenomic contigs by coverage and composition. Nat Methods. 2014;11:1144–6.
5. Lu YY, et al. COCACOLA: binning metagenomic contigs using sequence Composition, read Coverage, CO-alignment, and paired-end read Linkage. Bioinformatics. 2017;33(6):791–8.
6. Huson DH, et al. MEGAN analysis of metagenomic data. Genome Res. 2007;17(3):377–86.
7. Wood DE, Salzberg SL. Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol. 2014;15(3):R46.
8. Finn RD, et al. The Pfam protein families database towards a more sustainable future. Nucleic Acids Res. 2016;44(D1):D279–85.
9. Rosen GL, Reichenberger ER, Rosenfeld AM. NBC: the naive Bayes classification tool webserver for taxonomic classification of metagenomic reads. Bioinformatics. 2011;27(1):127–9.
10. Klyuyk A, et al. Unsupervised statistical clustering of environmental shotgun sequences. BMC Bioinformatics. 2009;10(1):316.
11. Kelley DR, Salzberg SL. Clustering metagenomic sequences with interpolated Markov models. BMC Bioinformatics. 2010;11(1):544.
12. Stouros M, et al. The binning of metagenomic contigs for microbial physiology of mixed cultures. Front Microbiol. 2012;3:410.
13. Laczy CC, et al. VizBin-an application for reference-independent visualization and human-augmented binning of metagenomic data. Microbiome. 2015;3(1):1.
14. Leung HC, et al. A robust and accurate binning algorithm for metagenomic sequences with arbitrary species abundance ratio. Bioinformatics. 2011;27(1):114–9.
15. Wu Y-W, Ye Y. A novel abundance-based algorithm for binning metagenomic sequences using i- tuples. J Comput Biol. 2011;18(3):523–34.
16. Imelfort M, et al. GroupM: an automated tool for the recovery of population genomes from related metagenomes. PeerJ. 2014;2:e603.
17. Wu Y-W, et al. MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. Microbiome. 2014;2(1):26.
18. Wu Y-W, Simmons BA, Singer SW. MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. Bioinformatics. 2016;32(4):605–7.
19. Wang Y, Hu H, Li X. MBBC: an efficient approach for metagenomic binning based on clustering. BMC Bioinformatics. 2015;16(1):36.
20. Lin H-H, Liao Y-C. Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes. Sci Rep. 2016;6:24175.
21. Karlin S, Mazeik J, Campbell AM. Compositional biases of bacterial genomes and evolutionary implications. J Bacteriol. 1997;179(12):3899–913.
22. Dick GJ, et al. Community-wide analysis of microbial genome sequence signatures. Genome Biol. 2009;10(8):R85.
23. Wan L, et al. Alignment-free sequence comparison (I): theoretical power of comparison statistics. J Comput Biol. 2010;17(11):1467–90.
24. Ahlgren NA, et al. Alignment-free d_{oligo} frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences. Nucleic Acids Res. 2017;45(1):39–53.
25. Song K, et al. Alignment-free sequence comparison based on next-generation sequencing reads. J Comput Biol. 2013;20(2):64–79.
26. Wang Y, et al. Comparison of metatranscriptomic samples based on k-tuple frequencies. PLoS One. 2014;9(11):e84348.
27. Liao W, et al. Alignment-free transcriptomic and Metatranscriptomic comparison using sequencing signatures with variable length Markov chains. Sci Rep. 2016;6:37243.
28. Jiang B, et al. Comparison of metagenomic samples using sequence signatures. BMC Genomics. 2012;13(1):730.
29. Wang Y, et al. MetaCluster 4.0: a novel binning algorithm for NGS reads and huge number of species. J Comput Biol. 2012;19(2):241–9.
30. Wang Y, et al. MetaCluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. Bioinformatics. 2012;28(18):i356–62.
31. Richter DC, et al. MetaSim—a sequencing simulator for genomics and metagenomics. PLoS One. 2008;3(10):e3373.
32. Zerbino DR, Birney E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res. 2008;18(5):821–9.
33. Mavromatis K, et al. Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. Nat Methods. 2007;4(6):495–500.
34. Hallam SJ, et al. Genomic analysis of the uncultivated marine crenarchaeote Cenarchaeum symbiosum. Proc Natl Acad Sci. 2006;103(48):18296–301.
35. Tyson GW, et al. Community structure and metabolism through reconstruction of microbial genomes from the environment. Nature. 2004;428(6978):37–43.
36. Woyke T, et al. Symbiosis insights through metagenomic analysis of a microbial consortium. Nature. 2006;443(7114):950–5.
37. Tringe SG, et al. Comparative metagenomics of microbial communities. Science. 2005;308(5721):554–7.
38. Sharon I, et al. Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization. Genome Res. 2013;23(1):111–20.
39. Ijaz, U, Quince C. TAXAassign v0.4. https://github.com/umerijaz/taxaassign 2013.
40. Parks DH, et al. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res. 2015;25(7):1043–55.
41. Ball GH, Hall DJ. ISOQAT, a novel method of data analysis and pattern classification. Merlo Park, CA: Stanford research inst; 1965.
42. Wu Y-W, et al. MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. 2014 13 Apr 2017; Available from: http://downloads.jbei.org/data/microbial_communities/MaxBin/MaxBin.html.