An image processing method for metagenomic binning: multi-resolution genomic binary patterns

Samaneh Kouchaki[1,*], **Avraam Tapinos**[1], **and David L Robertson**[1,2,*]

[1]Evolution and Genomic Sciences; School of Biological Sciences; Faculty of Biology, Medicine and Health; The University of Manchester; Manchester; M13 9PT; UK.
[2]MRC-University of Glasgow Centre for Virus Research; Glasgow; G61 1QH; UK.
[*]samaneh.kouchaki@manchester.ac.uk; david.l.robertson@glasgow.ac.uk

ABSTRACT

Bioinformatics methods typically use textual representations of genetic information, represented computationally as strings or sub-strings of the characters A, T, G and C. Image processing methods offer a rich source of alternative descriptors as they are designed to work in the presence of noisy data without the need for exact matching. We introduce a method, multi-resolution local binary patterns (MLBP) from image processing to extract local 'texture' changes from nucleotide sequence data. We apply this feature space to the alignment-free binning of metagenomic data. The effectiveness of MLBP is demonstrated using both simulated and real human gut microbial communities. The intuition behind our method is the MLBP feature vectors permit sequence comparisons without the need for explicit pairwise matching. Sequence reads or contigs can then be represented as vectors and their 'texture' compared efficiently using state-of-the-art machine learning algorithms to perform dimensionality reduction to capture eigengenome information and perform clustering (here using RSVD and BH-tSNE). We demonstrate this approach outperforms existing methods based on k-mer frequency. The image processing method, MLBP, thus offers a viable alternative feature space to textual representations of sequence data. The source code for our Multi-resolution Genomic Binary Patterns method can be found at https://github.com/skouchaki/MrGBP.

Introduction

Algorithms in bioinformatics use 4 of genetic information, sequences of the characters A, T, G and C represented computationally as strings or sub-strings. For example, in genome assembly, exact substring matching of short k-mers of fixed length are typically used to identify related sequences/strings[1,2]. Although this approach works well for closely related data, it will fail predictably with divergent sequences, e.g. viruses, due to a lack of homologous regions retaining sufficient sequence identity for exact matching. While there are approaches that permit relaxed k-mer matching[3,4], the signal processing methods used in image processing offer an alternative feature space because they are designed to be rotation and scale invariant, and are generally less sensitive to noise by mapping data to a less detailed representation, i.e., 'texture' changes. Due to discriminative power and computational simplicity of such techniques, they have found applications to many areas[5]. Consequently, they may work better for divergent genome information such as found in microbial communities and in particular for viruses many of which remain uncharacterised.

Here our aim is to implement and test the image processing method, local binary patterns (LBP), for the extraction of local changes in numerical representations of genetic sequence data (Figure 1). LBP is a feature descriptor capturing local texture changes first introduced for segmenting an image in two-dimensions into several meaningful partitions[6,7]. It is based on assigning a code to each local window. Its one-dimensional implementation also found application to other signal processing areas including speech processing[8,9]. Moreover, LBP has a multi-resolution version, called multi-resolution LBP (MLBP), which considers texture changes at various scales[10]. We make the assumption that each genomic contig or sequence read has 'texture' patterns at various scales that can be extracted using MLBP. Furthermore, the arbitrary location of each pattern does not affect the extracted feature vector.

To demonstrate an application of the MBLP feature vector and its effectiveness, we consider the problem of unsupervised grouping reconstructed genomic contigs into species-level groups ('binning'). High-throughput (so-called next-generation) sequencing technologies have generated enormous volumes of data in metagenomic studies. In these samples, the sequence reads can be from the same or different genomes from a microbial community of viruses and bacteria, including divergent variants of the same species. Hence, reconstructing (assembling) individual genomes from this mixed data can be problematic. Moreover, sequencing errors, sequence repetition, insufficient coverage, and genetic diversity can give rise to fragmented assemblies. Furthermore, comparing metagenomic data to existing reference genomes (taxonomic binning) will only identify some of the reads/contigs present. Consequently, genome composition-based techniques[11,12] have been introduced as an

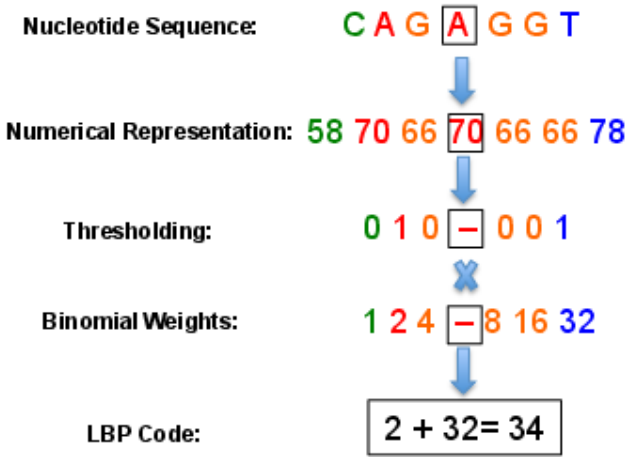

Figure 1. Calculating the LBP code. A threshold of the atomic numerical representation of the sequence (see Table 1) is determined by comparing the centre point (in the square) and its neighbours. The LBP code is then obtained by using binomial weights.

Table 1. The numerical value of each letter considering EIIP, atomic, real, and integer nucleotide representations.

Letter	EIIP	Atomic	Real	Integer
A	0.1260	70	-1.5	2
C	0.1340	58	-0.5	-1
G	0.0806	66	0.5	1
T	0.1335	78	1.5	-2

alternative way to analyse the species composition of metagenomic samples[13]. These methods use species-specific genomic signatures extracted by calculating the normalised frequency of k-mers of a specific size, e.g., commonly $k = 4$[14,15]. The signatures are obtained by counting the occurrences of each k-mer combination where the k-mer frequency of each sequence represents a feature vector in high-dimensional space.

A number of metagenomic binning techniques have employed genomic signatures. Across-sample coverage-profiles, or a hybrid approach, using genomic signatures is a commonly used feature[16,17]. For example, emergent self organising maps (ESOM) based binning uses contour boundaries to visualise the clusters[17]. Unfortunately, ESOM plots are computationally very demanding. Other methods that consider coverage across multiple samples include CONCOCT[16] and MetaBAT[18], however they require a high number of samples to perform well, e.g., 50 or more. VizBin[15] is another visualisation approach that considers a single sample, but it needs manual selecting of the centroids for binning.

To perform the clustering/binning we have first used singular value decomposition (SVD)[19,20] (specifically randomised SVD, RSVD[21] for time efficiency) to first reduce the dimensionality of the data, i.e., to identify the principle components of the MBLP feature vectors (termed 'eigengenome' information[22]). Second, these eigengenome features are passed as an input to Barnes-Hut t-distributed stochastic neighbor embedding (BH-tSNE)[23] for further dimensionality reduction and visualisation of the clusters in the data.

An overview of our approach Multi-resolution Genomic Binary Patterns (MrGBP) is depicted in Figure 2. We apply our method to both simulated and real metagenomic datasets, and demonstrate our results compare favourably to several existing binning methods. We also consider the effect of including coverage information across-samples in a hybrid approach to maximise the performance in longitudinal metagenomic samples, and demonstrate improved performance. Collectively our results demonstrate the use of an image processing method in bioinformatics, a new feature space for sequence analysis.

Results and Discussion

Calculating MLBP requires numerical data as an input (Figure 1). Thus, genomic sequences need to be first mapped into one or several numerical representations[24,25]. A group of such representation methods are based on biochemical or biophysical properties of DNA molecules while others are arbitrarily assigned numbers (Table 1). MLBP features can then be extracted

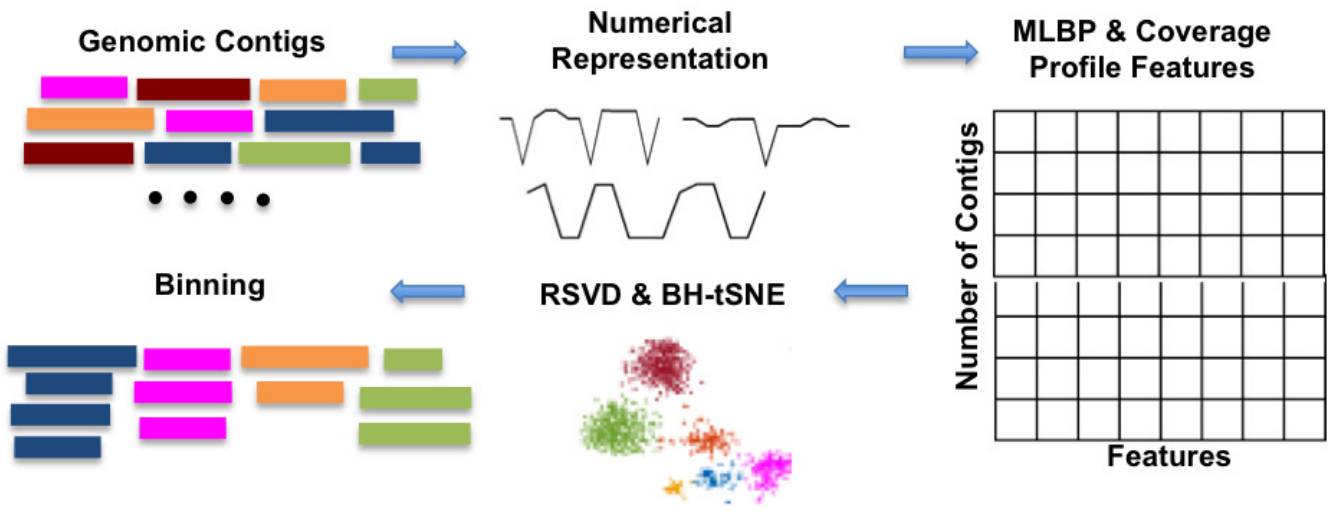

Figure 2. Schematic overview of our implementation of the MrGBP method to characterise the species relationships among metagenomic contigs.

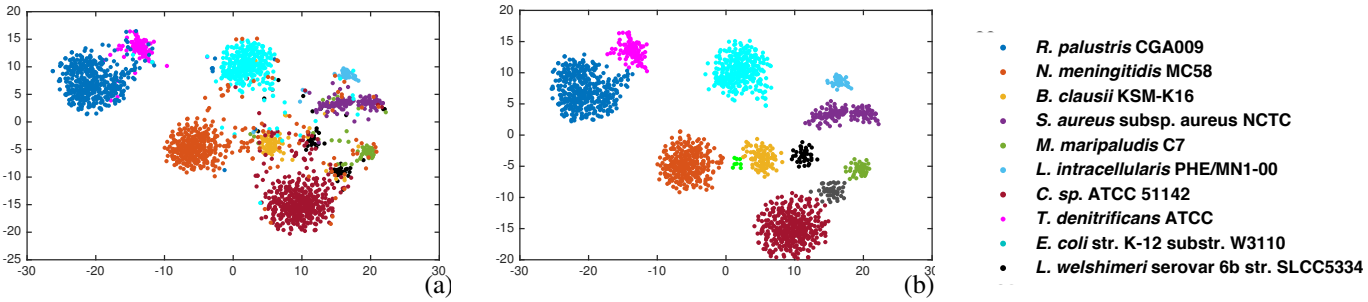

Figure 3. Visualisation of the simulated metagenomic community using Integer nucleotide mapping, MLBP to extract features, RSVD for feature reduction, BH-tSNE two-dimensional representation and cluster identification using DBSCAN comparing (a) manually annotated clusters (see species names in key) to (b) the DBSCAN defined clusters.

from these numerical representations and used to compare sequence data.

The performance of our method is tested for a low complexity simulated dataset using different numerical mappings (EIIP, atomic, real, and integer nucleotide representations, Table 1) for MLBP lengths $p \leq 6$ (supplementary Figure 1). For example, for the integer representation our automated binning approach very closely matches the manually annotated clusters (compare panels a and b in Figure 3). Specifically, the contigs from different species form visually separate clusters with very limited overlap with the clusters of other species.

The different numerical representations provided slightly different data clusters but overall the results demonstrated similar performance (Table 2). The Integer representation was selected for subsequent analysis as it has relatively high performance and more discrimination compared to the other representations. The average run time was 75.35 seconds (2184 contigs with total length 33138556 nucleotides). The run time includes loading the data, numerically representing the data, MLBP feature extraction, and dimension reduction using BH-tSNE (Figure 2).

To check the effect of changing the window length, we considered various lengths of MLBP windows (Table 3 and supplementary Figure 2). As the MLBP vectors are based on a histogram, the number of features is determined by the window length, which may effect final performance. Here, run time only includes the time to numerically represent the data and MLBP feature selection.

With smaller window lengths, the resulting feature vectors cannot describe the underlying structure of the metagenomic dataset, while larger feature vectors increases the time complexity (Table 3). Hence, window size should be sufficiently large to maintain the distinctness of the signal (information regarding texture changes across various contigs).

The computational complexity of our method increases as the dimensions of the feature space increase. Therefore, we considered how keeping different numbers of eigen factors can effect the performance and run time of our method (Figure 4).

Table 2. Precision, recall, F1 score (%), and the number of clusters for various nucleotide mappings for a simulated low complexity metagenomic dataset.

Nucleotide mapping	Atomic	EIIP	Real	Integer
Precision	97.23	89.08	97.41	98.38
Recall	94.82	96.80	93.96	96.35
F1 score	96.01	92.78	95.65	97.35
Number of clusters	12	10	13	12

Table 3. Precision, recall, F1 score (%), the number of clusters, and the run time(s) for MLBP of various window length or feature dimensions ($p \leq P$) and integer representation.

Window length	2	4	6	8
Feature dimension	4	20	84	212
Precision	86.08	97.78	98.38	98.21
Recall	85.80	94.04	96.35	94.63
F1 score	85.94	95.87	97.35	96.39
Number of clusters	19	16	12	11
Run time	22.43	27.89	36.22	46.14

Here, the numerical integer representation is considered for the nucleotide mapping and $p \leq 6$ for feature selection. The results show that after keeping a number of eigen factors, i.e., 30, the final performance does not change significantly. However, as the number of eigen factors increases the run time of RSVD and BH-tSNE increases (Table 4). These results show that the MLBP method can analyse a small metagenomic data in a short time. Moreover, it is performing well considering only one sample was analysed.

Comparison with Existing Methods for Simulated 10 and 100 Metagenomic Data

Here, we considered two simulated datasets with 10 and 100 genomes to compare our results to both low and complex metagenomic communities. Our results compared favourably with CONCOCT[16], MetaBAT[18], and MaxBin2[26,27] (Table 5). CONCOCT bins the data by employing sequence composition and across-sample coverage. The method has been compared with a range of methods including MetaWatt[28], SCIMM[29], and CompostBin[30] to show its advantage over composition based techniques. However, for high complexity metagenomic data CONCOCT will not work as well as other techniques. MetaBAT bins the metagenomic data using probabilistic distances of genome abundance with sequence composition. It is an efficient method for analysing complex metagenomic data. MaxBin was originally introduced for single sample data in which it bins the data based on tetra-nucleotides frequencies and it has been extended to MaxBin2 to support multiple samples. MetaBAT and MaxBin2 produce many unclassified contigs. Consequently, they have higher precision but lower recalls.

For the 100 simulated genomes data our method performs better than CONCOCT and MetaBin methods and quite close

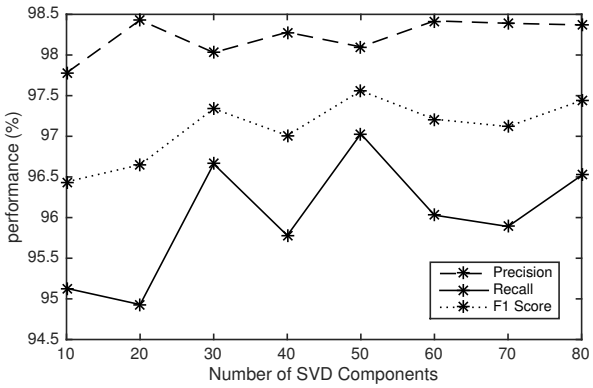

Figure 4. Precision, recall, and F1 score (%) by keeping different numbers of RSVD components. Integer representation and $p \leq 6$ have been considered to analyse the simulated metagenomic dataset.

Table 4. Run time(s) of RVSD and BH-tSNE for various number of RSVD components.

Number of RSVD Components	10	20	30	40	50	60	70	80
RSVD run time	0.06	0.12	0.20	0.30	0.44	0.55	0.69	0.85
BH-tSNE run time	27.76	28.83	30.04	31.20	33.05	34.65	35.63	35.80

Table 5. Precision, recall, F1 score (%), and the number of clusters for our proposed method, CONCOCT, MetaBAT, and MaxBin.

10 Genomes				
Methods	Precision	Recall	F1 score	Number of clusters
MLBP	98.38	96.35	97.36	12
CONCOCT	98.56	97.35	97.95	19
MetaBAT	90.82	94.98	92.85	9
MaxBin	93.43	96.65	95.01	10
100 Genomes				
Methods	Precision	Recall	F1 score	Number of clusters
MLBP	91.52	83.97	87.58	116
CONCOCT	60.73	96.37	74.51	79
MetaBAT	92.34	89.62	90.96	104
MaxBin	89.83	83.96	86.80	85

to MetaBAT. Lower performance is mainly because DBSCAN does not work very well for a very dense feature space (high complexity data representation). It may result in some unclustered contigs and therefore, lower performance. It still shows that the proposed pipeline can work for low and high complexity dataset. However, better clustering methods could improve the final results.

Real Data: Infant Human Gut

A relatively low-complexity infant human gut dataset[17] was analysed to test the performance of our method with real data. A main reason for considering this dataset is to show the effectiveness of the MBLP method to bin low abundant viral community where texture analysis should perform better. The integer numerical representation was used for the nucleotide mapping, $p \leq 8$ for feature selection, and the first 60 eigen components in the dimension reduction stage (RSVD).

The proposed method binned the data into 19 clusters with precision and recall of 88.34 and 97.22 at the species level. BH-tSNE representation of the data demonstrates the genomic contigs of the same or very similar contigs are binned together (Figure 5). While some of the plasmids and viruses (bacteriophages) clustered with associated clusters, most species formed their own cluster. The bacterial species tend to form separate clusters, for example, *Anaerococcus sp.* and *C. albicans* form clusters 1 and 3 (Figure 5). However, separating plasmid or virus from its host is less straight-forward due to their closer genome compositions. Nonetheless, our method manages to bin *S. aureus* strains, their plasmid, and virus into two groups; (1) *S. aureus* strain and plasmid and (2) *S. aureus* strain 2 and virus. *Propionibacterium sp.* appears as a separate bin. *E. faecalis* and one of its plasmids forms one cluster. *S. epidermidis* has three strains, three viruses, one integrated virus (prophage) and several plasmids, and the algorithm managed to bin them into five clusters where *S. epidermidis* strains 1 and 3 clustered together (including virus 13 and 14), with strain 4 forming a separate cluster (including virus 46).

Our results compare favorably with CONCOCT[16], MetaBAT[18], and MaxBin2[26,27]. The results show better performance on this dataset with small sample size (11 samples) in comparison with the other techniques (Table 6).

To further investigate the relationship between clusters, the abundance patterns of each cluster were calculated based on the number of reads mapped to contigs at the different sampling time points (Supplementary Figure 3). Pairwise correlation coefficients were then calculated to check for any pattern among the clusters. The results suggests that there is a strong correlation between clusters of related species (Supplementary Figure 3). For example, the clusters of *Propionibacterium* and *Peptoniphilus* species have similar abundance patterns (Clusters 9-10). Similar results were also found in[17] where both species have proliferation in later stages and hence are well-adapted to the gut. Moreover, two clusters has been formed for *F. magna* with very similar coverage patterns (clusters 5-6). Consequently, this similarity could be analysed further to join some of the clusters. A similar pattern can be observed in the clusters of *S. aureus*, confirming the relationship between each bacteria and virus (clusters 11-12). The five clusters of *S. epidermidis* also share similar coverage patterns (clusters 13-17). A further step could be to cluster all the contigs of these five clusters separately to have a better separation of the related strains and viruses.

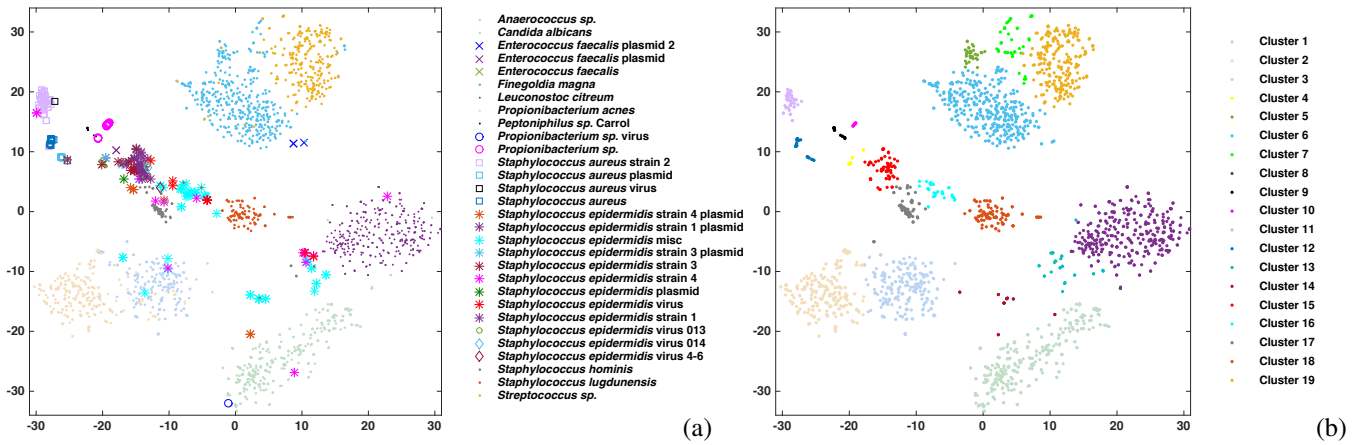

Figure 5. Visualisation of the infant gut metagenomic community using integer nucleotide mapping, MLBP to extract features, RSVD feature reduction, BH-tSNE two-dimensional representation and cluster identification using DBSCAN comparing (a) manually annotated clusters (see bacteria species, virus or plasmid names in key) to (b) the DBSCAN defined clusters 1 to 19.

Table 6. Precision, recall, F1 score (%), and the number of clusters for our proposed method, CONCOCT, MetaBAT, and MaxBin2.

Methods	Precision	Recall	F1 score	Number of clusters
MLBP	88.34	97.22	92.57	19
CONCOCT	79.5	90.62	84.69	32
MetaBAT	84.23	92.35	88.10	10
MaxBin2	82.84	93.50	87.84	10

Finally, we checked the run time of our method. It takes about three minutes (108.67 s) to analyse this dataset (the number of contigs is 2293 and total length of them is 27594702). Although our code is relatively fast, it could be further optimised in terms of both time and memory.

Conclusion

We have demonstrated that the image processing technique, MBLP, can be applied to numerical nucleotide sequence data comparisons. Specifically, a metagenomic visualisation and binning approach has been implemented by representing the nucleotide genomic contigs numerically. MLBP was employed to capture the genomic signature changes followed by dimensionality reduction steps to visualise the data in a lower dimension. Our results on simulated genomic fragments show the underlying taxonomic structure of the metagenomic data and verify the advantage of using signal processing approaches for metagenomic data analysis. The results illustrates that our method can be used for the visualisation and clustering of human gut metagenomic data at the genus or species level. In addition, only a limited number of contigs overlap with the clusters of other species.

Methods

Our methodological pipeline (Figure 2) is comprised of several steps: We first numerically represent the genomic contigs using a nucleotide mapping (Table 1). MLBP is then used to extract features from these numerical representations. If available, cross-sample coverage information (mean and standard deviation) is extracted separately using Bowtie2[31],and can be considered as extra information to be added in the MLBP feature space. RSVD is used to reduce the dimensions of the MLBP feature vectors by capturing the eigengenome information. BH-tSNE is then used to map RSVD features to a two-dimensional space for visualisation and data binning. For quantitatively evaluating the visualisation performance, we cluster the BH-tSNE projected data using DBSCAN a density-based spatial clustering algorithm[32] and calculate the precision, recall, and F1 score between the DBSCAN assigned labels and the original labels.

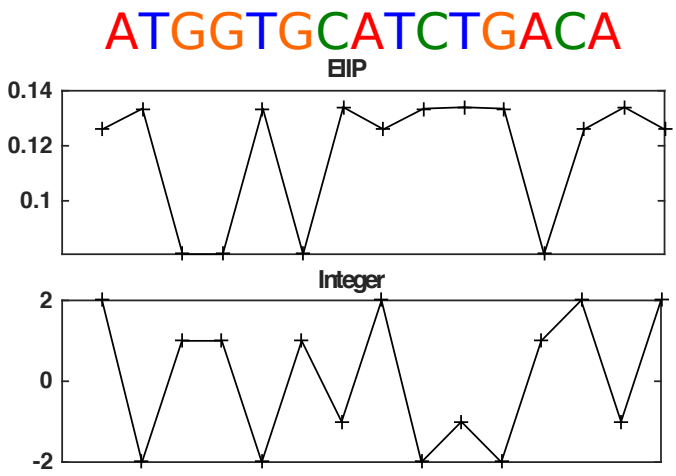

Figure 6. A nucleotide sequence (top) and example representations: EIIP and integer. Each nucleotide A, C, G, or T in the sequence is assigned to a value depending on the numerical representation.

The Nucleotide Mapping

Methods to numerically represent the genomic reads can be categorised into two groups: (i) Assigning an arbitrary value to each letter A, C, G, or T of the nucleotide sequence: Voss[33], two or four bit binary representations[34,35] can be considered as examples of this group. (ii) Defining numerical representations that correspond to certain biochemical or biophysical properties of the DNA molecules: electron ion interaction potential (EIIP)[36], paired nucleotide representations[37], and atomic representations[38] are examples of this group.

Various representations carry different properties (texture patterns) of each sequence. Here, we compare the EIIP, atomic, real, and integer nucleotide representations. Table 1 shows the value assigned to each nucleotide in each of the representations. Figure 6 shows an example of mapping a nucleotide sequence to two numerical vectors.

Multi-resolution Local Binary Patterns

Using LBP, each two-dimensional window is mapped to a binary number with a fixed length. LBP codes illustrate the data patterns (e.g., for textural changes in images and frequency changes in speech), while the histogram distribution shows how often each pattern appears. These histograms are considered as the feature vectors which essentially extract the species specific genomic signatures.

LBP assigns a binary code to each sample by examining its neighbouring points. By considering $x(t)$ as the numerical representation of the tth position of genomic segment, LBP is defined as

$$\begin{aligned}\mathrm{LBP}(x(t)) = \sum_{i=0}^{p/2-1} \{&\mathrm{Sign}(x(t+i-p/2)-x(t))2^i + \\ &\mathrm{Sign}(x(t+i+1)-x(t))2^{i+p/2}\},\end{aligned} \tag{1}$$

where p is the number of neighbouring points and Sign indicates the sign function

$$\mathrm{Sign}(x) = \begin{cases} 0 & x<0 \\ 1 & x\geq 0 \end{cases}. \tag{2}$$

Sign assigns a binary number by thresholding the difference between each neighbouring point and the centre point t. Consequently, it assigns a p-bit binary number to each window of length $p+1$. Each binary number is converted to a LBP code using a binomial weight. An example of the LBP operator can be seen in Figure 1 where $p=6$. The value of the centred point (in the square in Figure 1) is compared with the six neighbouring points to produce the LBP code. This code describes the data changes locally all in a compressed format. Finally, by considering all the obtained codes, the distribution of the LBP codes can be defined as

$$\mathbf{h}_k = \sum_{p/2\leq i\leq N-p/2} \delta(\mathrm{LBP}_p(x(i),k)), \tag{3}$$

where $k = 1, 2, ..., 2^p$ is all possible values of LBP codes, δ shows the Kronecker delta function, and N is the genomic fragment length. Considering the distribution of LBP codes makes the feature space independent of each pattern location and only dependent to frequency of each MLBP code.

MLBP is an LBP extension that combines the results of LBP distribution from various values of $p \leq P$. Consequently, the pattern changes of different resolution levels are considered to improve the description of the data inputs. Here, we apply MLBP to one-dimensional linear sequences to consider pattern changes of various lengths.

Across-Samples Coverage Information

To obtain the coverage profile for contigs across the longitudinal samples, the Illumina reads were mapped to contigs with Bowtie2[31] for each time point. Samtools[39,40] was then used to produce a per base depth file. As a result, our coverage feature vector for each genomic contig is the average and standard deviation of the per base depth for each contig. Coverage information provides extra information that optionally can be added to the MLBP feature space.

Randomised Singular Value Decomposition

A metagenomic community can be considered as a linear combination of genomic variables. The histogram of MLBP codes for each genomic fragment captures the changes in the pattern (the "texture") of each distinct fragment. By representing a vector of MLBP codes for each fragment, low-rank matrix approximations can be used for efficient analysis of the metagenomic data. Our assumption in using SVD is that the MLBP codes of the contigs from each species have a distinct energy contribution. Therefore, the data can be represented as a linear combination of mutually independent components.

SVD decomposition of a matrix $\mathbf{X}$ is defined as

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{4}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors, Σ is singular values, and $(\cdot)^T$ denotes the transpose operator.

SVD can be time consuming when dealing with large scale problems such as metagenomic data analysis. Therefore, RSVD is used as an accurate and robust solution to estimate the dominant eigen components quickly[41].

RSVD calculates the first ith eigen components of the data by using QR decomposition and mapping $\mathbf{X}$ to a smaller matrix as

$$\begin{aligned} &\Omega = \text{randn}(N, i), \\ &\mathbf{Y} = \mathbf{X}\Omega, \mathbf{Y} = \mathbf{QR} \\ &\mathbf{B} = \mathbf{Q}^T\mathbf{X}, \end{aligned} \tag{5}$$

where randn generates a random matrix of the size of its inputs and N is the number of contigs. After decomposing $\mathbf{B}$ using SVD, the final factors are obtained using $\mathbf{Q}$ and the eigen factors of $\mathbf{B}$.

Barnes-Hut t-Distributed Stochastic Neighbor Embedding

BH-tSNE has become a common technique for high-dimensional data visualisation in several applications[23]. It is based on the divergence minimisation of two distributions: pairwise similarities of the input objects and the corresponding low-dimensional points. As a result, the data in the final lower dimension keeps the original local data structure.

The ordinary similarity measure of the data points is defined based on normalised Gaussian kernel values that scales quadratically to the number of data points. The main objective function is approximated by defining the similarity function based on a number of neighbouring points[23]. In addition, a vantage-point tree is employed for rapidly finding the neighbouring points. BH-tSNE is thus a more efficient ($O(N\log N)$) data reduction approach and used in this paper for data visualisation and binning. The data dimension is usually reduced to two or three dimensions. Our suggestion is to use three for complex and two for low complexity data.

Datasets

To validate the effectiveness of our methodology we consider both simulated and real datasets. Simulated metagenomic data of Illumina sequences for 10 and 100 genomes (supplementary Tables 1 and 3) was downloaded from http://www.bork.embl.de/~mende/simulated_data/. The data were assembled by Ray Meta[42] into contigs ($k = 31$). Using these datasets, various aspects of our method, including MLBP window length and RSVD number of eigen components, has been analysed.

For the real data analysis, a time-series metagenomics human gut dataset comprised of 11 samples (18 runs) taken over nine days from a newborn infant[17] was analysed. The authors have assembled the data into 2329 contigs. This assembly and binning information is provided at http://ggkbase.berkeley.edu/carrol/. Corresponding Illumina reads can be downloaded from the NCBI, SRA052203, which consists of 18 Illumina sequencing runs (SRR492065-66 and SRR492182-97). For the real data, we mapped the reads to the contigs using Bowtie2 and coverage profiles have been obtained using SAMtools.

Performance Evaluation

In order to check the performance of our MrGBP method, DBSCAN[32] has been used to cluster the final results. The precision, recall, and F1 score are calculated between the DBSCAN assigned labels and the original labels to determine the performance as a measure of a clusters "purity". Assuming there are m genomes in the dataset and it is binned to k clusters, the precision, recall, and F1 score can be calculated as

$$\begin{aligned} \text{Precision} &= \frac{\sum_{i=1}^{k} \max_{j} s_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{m} s_{ij}} \\ \text{Recall} &= \frac{\sum_{j=1}^{m} \max_{i} s_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{m} s_{ij} + \sum \text{unbinned sequences}} \\ \text{F1} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \tag{6}$$

where s_{ij} is the length of contigs in cluster i corresponds to genome j.

Selecting DBSCAN parameters

DBSCAN does not need the number of clusters but has two parameters that need to be determined: epsilon that indicates the closeness of the points of each cluster to each other and minPts, the minimum neighbours a point should have to be considered into a cluster. Usually these values are not known prior to analysis and there are several ways to select their values. One way is to calculate the distance of each point to its closest nearest neighbour and use the histogram of distances to select epsilon. After selecting epsilon a histogram can be obtained of the average number of neighbours for each point using the epsilon. Some of the samples do not have enough neighbouring points and can be considered as noise. Implementation of the parameter selection is included in spark_dbscal (https://github.com/alitouka/spark_dbscan). DBSCAN may result in some unclustered samples.

References

1. Zerbino, D. R. & Birney, E. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome research* **18**, 821–829 (2008).
2. Bankevich, A. *et al.* SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of computational biology* **19**, 455–477 (2012).
3. Healy, J. & Chambers, D. Approximate k-mer matching using fuzzy hash maps. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **11**, 258–264 (2014).
4. Shajii, A., Yorukoglu, D., Yu, Y. W. & Berger, B. Fast genotyping of known SNPs through approximate k-mer matching. *Bioinformatics* **32**, i538–i544 (2016).
5. Zhao, Y. Theories and applications of LBP: a survey. In *International Conference on Intelligent Computing*, 112–120 (Springer, 2011).
6. Pietikäinen, M. & Ojala, T. Texture analysis in industrial applications. In *Image Technology*, 337–359 (Springer, 1996).
7. Tirunagari, S. *et al.* Detection of face spoofing using visual dynamics. *IEEE Transactions on Information Forensics and Security* **10**, 762–777 (2015).
8. Chatlani, N. & Soraghan, J. J. Local binary patterns for 1-D signal processing. In *Signal Processing Conference, 2010 18th European*, 95–99 (IEEE, 2010).
9. Alegre, F., Amehraye, A. & Evans, N. A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, 1–8 (IEEE, 2013).
10. Ojala, T., Pietikainen, M. & Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence* **24**, 971–987 (2002).

11. Blaisdell, B. E. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences* **83**, 5155–5159 (1986).

12. Blaisdell, B. E. Markov chain analysis finds a significant influence of neighboring bases on the occurrence of a base in eucaryotic nuclear DNA sequences both protein-coding and noncoding. *Journal of molecular evolution* **21**, 278–288 (1985).

13. Mande, S. S., Mohammed, M. H. & Ghosh, T. S. Classification of metagenomic sequences: methods and challenges. *Briefings in bioinformatics* 669–681 (2012).

14. Lin, H.-H. & Liao, Y.-C. Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes. *Scientific reports* **6** (2016).

15. Laczny, C. C. *et al.* VizBin-an application for reference-independent visualization and human-augmented binning of metagenomic data. *Microbiome* **3**, 1 (2015).

16. Alneberg, J. *et al.* Binning metagenomic contigs by coverage and composition. *Nature methods* **11**, 1144–1146 (2014).

17. Sharon, I. *et al.* Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization. *Genome research* **23**, 111–120 (2013).

18. Kang, D. D., Froula, J., Egan, R. & Wang, Z. MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. *PeerJ* **3**, e1165 (2015).

19. Golub, G. H. & Reinsch, C. Singular value decomposition and least squares solutions. *Numerische mathematik* **14**, 403–420 (1970).

20. Paukkeri, M.-S., Kivimäki, I., Tirunagari, S., Oja, E. & Honkela, T. Effect of dimensionality reduction on different distance measures in document clustering. In *International Conference on Neural Information Processing*, 167–176 (Springer, 2011).

21. Halko, N., Martinsson, P.-G. & Tropp, J. A. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions (2009).

22. Cleary, B. *et al.* Detection of low-abundance bacterial strains in metagenomic datasets by eigengenome partitioning. *Nature biotechnology* **33**, 1053–1060 (2015).

23. Van Der Maaten, L. Accelerating t-SNE using tree-based algorithms. *Journal of machine learning research* **15**, 3221–3245 (2014).

24. Lorenzo-Ginori, J. V., Rodríguez-Fuentes, A., Abalo, R. G. & Rodriguez, R. S. Digital signal processing in the analysis of genomic sequences. *Current Bioinformatics* **4**, 28–40 (2009).

25. Tapinos, A., Constantinides, B., Kelland, D. B. & Robertson, D. L. Alignment by the numbers: sequence assembly using reduced dimensionality numerical representations. *bioRxiv* 011940 (2014).

26. Wu, Y.-W., Tang, Y.-H., Tringe, S. G., Simmons, B. A. & Singer, S. W. MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. *Microbiome* **2**, 1 (2014).

27. Wu, Y.-W., Simmons, B. A. & Singer, S. W. Maxbin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics* 605–607 (2015).

28. Chatterji, S., Yamazaki, I., Bai, Z. & Eisen, J. A. CompostBin: A DNA composition-based algorithm for binning environmental shotgun reads. In *Annual International Conference on Research in Computational Molecular Biology*, 17–28 (Springer, 2008).

29. Kelley, D. R. & Salzberg, S. L. Clustering metagenomic sequences with interpolated markov models. *BMC bioinformatics* **11**, 1 (2010).

30. Kislyuk, A., Bhatnagar, S., Dushoff, J. & Weitz, J. S. Unsupervised statistical clustering of environmental shotgun sequences. *BMC bioinformatics* **10**, 1 (2009).

31. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with bowtie 2. *Nature methods* **9**, 357–359 (2012).

32. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, vol. 96, 226–231 (1996).

33. Voss, R. F. Evolution of long-range fractal correlations and 1/f noise in DNA base sequences. *Physical review letters* **68**, 3805 (1992).

34. Ranawana, R. & Palade, V. A neural network based multi-classifier system for gene identification in DNA sequences. *Neural Computing & Applications* **14**, 122–131 (2005).

35. Demeler, B. & Zhou, G. Neural network optimization for E. coli promoter prediction. *Nucleic acids research* **19**, 1593–1599 (1991).

36. Nair, A. S. & Sreenadhan, S. P. A coding measure scheme employing electron-ion interaction pseudopotential (EIIP). *Bioinformation* **1**, 197–202 (2006).

37. Bernaola-Galván, P., Carpena, P., Román-Roldán, R. & Oliver, J. Study of statistical correlations in DNA sequences. *Gene* **300**, 105–115 (2002).

38. Holden, T. *et al.* ATCG nucleotide fluctuation of Deinococcus radiodurans radiation genes. In *Optical Engineering+ Applications*, 669417–669417 (International Society for Optics and Photonics, 2007).

39. Li, H. *et al.* The sequence alignment/map format and samtools. *Bioinformatics* **25**, 2078–2079 (2009).

40. Li, H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* **27**, 2987–2993 (2011).

41. Halko, N., Martinsson, P.-G. & Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review* **53**, 217–288 (2011).

42. Boisvert, S., Raymond, F., Godzaridis, É., Laviolette, F. & Corbeil, J. Ray Meta: scalable de novo metagenome assembly and profiling. *Genome biology* **13**, 1 (2012).

Acknowledgements

SK is supported by the VIROGENESIS project. The VIROGENESIS project receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 634650. AT is supported by BBSRC project grant, BB/M001121/1. We would like to thank Bede Constantinides for help with metagenomics data analysis and Santosh Tirunagari for helpful comments.

Author contributions statement

SK designed and wrote the methods and software and performed the data analysis. AT provided useful comments on the nucleotide mapping and software design. SK and DLR conceived the study. SK wrote the manuscript with comments from AT and DLR. All authors read and approved the final manuscript.

Additional information

The source code for our MrGBP method can be found at `https://github.com/skouchaki/MrGBP`. The code was tested on a Mac OS X operation system with 3.7 GHz Intel Xeon processor with 4 GB RAM. The code can be used to extract MLBP codes, visualise, and bin metagenomic contigs. Coverage information is an optional input file. Except for the input contig fasta file, all other parameters are set to default values but can be changed to values of interest.

The corresponding author is responsible for submitting a competing financial interests statement on behalf of all authors of the paper.