Metagenomic Binning through Multi-resolution Genomic Binary Patterns

Samaneh Kouchaki, Avraam Tapinos, David L Robertson
Computational and Evolutionary Biology, Faculty of Life Science, The University of Manchester
Email: samaneh.kouchaki@manchester.ac.uk

***Abstract*—Motivation: High-throughput sequencing has facilitated the analysis of complex microbial communities. Consequently, an enormous number of sequences have been generated containing various regions of bacterial and viral genomes. Image processing offers a rich source of descriptors for data analysis. Here, we introduce a feature space called multi-resolution local binary patterns (MLBP) from image processing as a feature descriptor to extract local 'texture' changes from nucleotide sequences. We demonstrate its applicability to the alignment-free binning of metagenomic data.**

Results: The effectiveness of our approach is tested using both simulated and real human gut microbial communities. We compared the performance of our method with several existing techniques that are based on k-mer frequency to show it outperforms existing techniques. In addition, we provide a time-series study of the abundance pattern of each bin to help refine the formed clusters automatically and to find relations that may exist among the clusters. Although the main aim is to introduce the use of genomic signatures using an alternative feature space (MLBP), our results show its application to the analysis of contigs from a metagenomic study.

Availability: The source code for our Multi-resolution Genomic Binary Patterns method can be found at https://github.com/skouchaki/MrGBP

I. Introduction

High-throughput (so-called next-generation) sequencing technologies are generating an enormous volume of biological data. Sequence analysis therefore plays an important role in studying the genetic information present in the sequenced samples. In metagenomic studies the sequence reads can be from the same or different genomes from a community of viruses and bacteria. Hence, reconstructing 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. Consequently, alignment-free techniques [1], [2] have been introduced as an alternative way for analysing the species composition of metagenomic data [3]. A main category of alignment-free techniques are based on species-specific genomic *signatures* extracted by calculating the normalised frequency of k-mers of a specific size, e.g., $k = 4$. 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.

Here our aim is to introduce an alternative feature space, local binary patterns (LBP), to extract the local changes in a sequence. LBP is a feature descriptor capturing local texture changes first introduced for segmenting an image into several meaningful partitions [4], [5]. Its one-dimensional implementation also found application to other signal processing areas including speech processing [6]. Moreover, it has a multi-resolution version, called multi-resolution LBP (MLBP), which considers texture changes at various scales [7]. Here, we assume that each genomic contig or sequence read has 'texture' patterns at various scales that can be extracted using MLBP. Moreover, the arbitrary location of each pattern does not affect the extracted feature vector. However, calculating MLBP requires numerical data as an input. Thus, genomic sequences need to be mapped into one or several numerical representations [8], [9]. A group of such representation methods are based on biochemical or biophysical properties of DNA molecules and some others are arbitrary assigned numbers. MLBP features can be extracted from these numerical representations, that can be used to analyse the metagenomic data.

To demonstrate an application of this feature vector and its effectiveness, we consider the problem of automatically grouping reconstructed genomic contigs into species-level groups ('binning'). Binning plays an important role in metagenomic analysis as it groups related reads or contigs for further analysis. Unsupervised binning and visualisation of the metagenomic data is especially helpful when there is no related reference genomes or any other prior information about the taxonomic structure of the data.

A number of metagenomic binning techniques have employed genomic signatures. Across-sample coverage-profiles, or a hybrid approach, using genomic signatures are commonly used to describe genomic fragments [10], [11]. Emergent self organising maps (ESOM) based binning is one example that uses contour boundaries to visualise the clusters [11]. Unfortunately, ESOM plots are computationally very expensive. There are also methods that consider coverage across multiple samples, e.g., CONCOCT [10] and MetaBAT [12], however it requires a high number of samples to perform well, e.g., 50 or more. VizBin [13] is another reference-independent visualization approach that considers a single sample, but it needs manual selecting of the centroids for binning.

Here, for extracting the features from the numerical mapping of nucleotide contigs, we use MLBP in one-dimension. We also consider the effect of considering coverage information across-samples in a hybrid approach to maximise the performance in longitudinal metagenomic samples. For visualisation, the feature vectors need to be projected from a higher di-

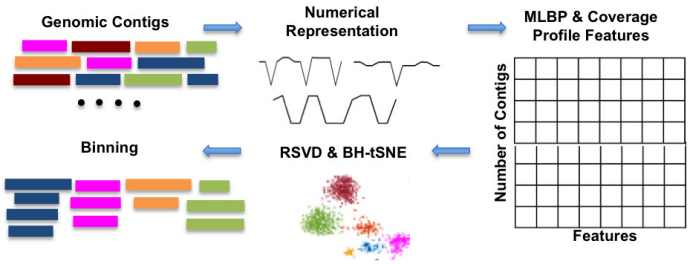

Fig. 1. Schematic overview of our implementation of the Multi-resolution Genomic Binary Patterns method to characterise the species relationships among metagenomic contigs.

mensional to lower dimensional space (e.g., two-dimensions). The common feature reduction techniques are: (i) linear based, such as singular value decomposition (SVD) [14], [15], and (2) non-linear based, such as ESOMs [16] and Barnes-Hut t-Distributed stochastic neighbor embedding (BH-tSNE) [17]. Although the non-linear techniques preserve the underlying structure of the data, they are computationally expensive. Therefore, we have used randomised SVD (RSVD) [18] to first reduce the higher dimension to obtain 'eigengenome' information [19] in a shorter time. Finally, there eigengenome features are passed as an input to BH-tSNE for visualising and binning the metagenomic dataset.

II. Methods

In this section we present our methodological pipeline (Figure 1). We numerically represent the genomic contigs using a nucleotide mapping (Table I). After that MLBP is used to extract features from these numerical representations. In addition, across-sample coverage information (mean and standard deviation) is extracted separately using Bowdtie2 [20]. 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. For quantitatively evaluating the visualisation performance, we cluster the BH-tSNE projected data using DBSCAN a density-based spatial clustering algorithm [21] and calculate the precision, recall, and F1 score between the DBSCAN assigned labels and the original labels.

A. The Nucleotide Mapping

Methods to numerically represent the genomic reads can be categorised into two groups: (1) Assigning an arbitrary value to each letter A, C, G, or T of the nucleotide sequence: Voss [22], two or four bit binary representations [23], [24] can be considered as examples of this group. (2) Defining numerical representations that correspond to certain biochemical or biophysical properties of the DNA molecules: electron ion interaction potential (EIIP) [25], paired nucleotide representations [26], and atomic representations [27] are examples of this group.

Various representations carry different properties (texture patterns) of each sequence. Here, we compare the EIIP, atomic, paired, real, and integer nucleotide representations. Table I shows the value assigned to each nucleotide in each of the representations. Figure 2 shows an example of mapping a nucleotide sequence to two numerical vectors.

TABLE I
The numerical value of each letter considering EIIP, atomic, paired, real, and integer nucleotide representations.

Letter	EIIP	Atomic	Paired	Real	Integer
A	0.1260	70	0	-1.5	2
C	0.1340	58	1	-0.5	-1
G	0.0806	66	1	0.5	1
T	0.1335	78	0	1.5	-2

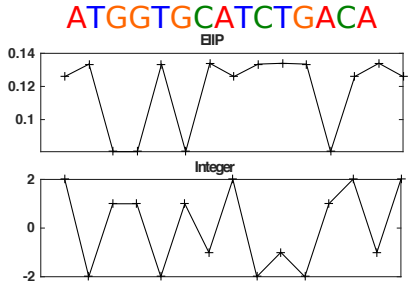

Fig. 2. A nucleotide sequence (top) and example representations: EIIP and integer. Each nucleotide A, C, G, or T in the sequence is assigned to a value depending on the numerical representation.

B. Multi-resolution Local Binary Patterns

LBP and its extensions (e.g., MLBP) have gained significant popularity in the field of image, speech, and signal processing [28]. 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 tth sample of the numerical representation of a genomic segment, LBP is defined as

$$\mathrm{LBP}(x(t)) = \sum_{i=0}^{p/2-1} \{\mathrm{Sign}(x(t+i-p/2)-x(t))2^i + \mathrm{Sign}(x(t+i+1)-x(t))2^{i+p/2}\}, \quad (1)$$

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

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

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 3 where $p = 6$. The value of the centred point (in the square in Figure 3) is compared with the

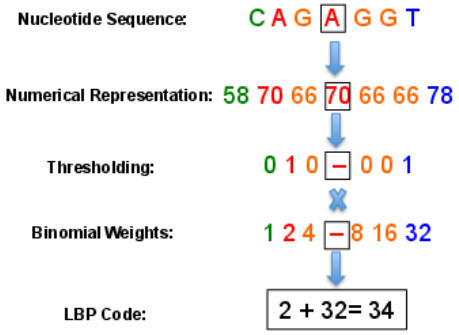

Fig. 3. Calculating the LBP code. A threshold of the atomic numerical representation of the sequence is determined by comparing the centre point (in the square) and its neighbours. The LBP code is then obtained by using binomial weights.

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

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

where $k = 1, 2, ..., 2^p$ and N is the genomic fragment length. Considering the distribution makes the feature space dependent of happening location of each pattern.

MLBP is an LBP extension that combines the results of LBP distribution from various values of $p \le P$. 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.

C. Across-Samples Coverage Information

To obtain the coverage profile for contigs across the longitudinal samples, the Illumina reads were mapped to contigs with Bowtie2 [20] for each time point. Samtools [29], [30] 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.

D. Randomised Singular Value Decomposition

A metagenomic community can be considered as a linear combination of genomic variables. The sequence 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 [31].

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

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

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 $\mathbf{B}$.

E. Barnes-Hut t-Distributed Stochastic Neighbor Embedding

BH-tSNE has become a common technique for high-dimensional data visualisation in several applications [17]. 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 [17]. 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.

F. Datasets

To validate the effectiveness of our methodology we consider both simulated and real datasets. Simulated metagenomic data of Illumina sequences for 10 genomes was downloaded from http://www.bork.embl.de/~mende/simulated_data/27. The data were assembled by Ray Meta [32] into contigs ($k = 31$). Their %GC and genome size are illustrated in supplementary Table 1. Using this dataset, various aspects of our method, including MLBP window length and RSVD number of eigen components, are analysed.

For the real data analysis, a time-series metagenomics human gut dataset comprised of 11 samples taken over nine days from faeces from a newborn infant [11] 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. For both datasets only contigs with a length equal or longer than 1000 bp has been considered.

G. Performance Evaluation

In order to check the performance of our Multi-resolution Genomic Binary Patterns method, DBSCAN [21] 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 is 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.

H. 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. Our implementation assigns the unclustered points to the closest cluster.

III. Results and Discussion

In this section we present our analysis and results on both real and simulated data, discuss the results, and provide the run time(s) using 2.8 GHz Intel Xeon processor with 4 GB RAM. First, various aspects of our proposed method are analysed and discussed, then, the real data is analysed and compared with some existing tools.

A. Effect of Selecting the Nucleotide Mapping

Since various representations assign different values to each letter, they can lead to different texture patterns. Consequently, it can affect the final binning performance and the data visualisation. This can be considered as an advantage of using numerical representations as it provides the option of having various feature spaces and representations that can be selected based on applications or the final results.

Here, the performance of our method is illustrated for different numerical mappings (EIIP, atomic, paired, real, and integer nucleotide representations) for MLBP lengths $p \leq 6$ (supplementary Figure 1, Figure 4, and Table II). The EIIP representation is selected for the following subsection as it has high performance and more discrimination compared to other representations. As demonstrated, all mapping methods, except for the paired number provide, slightly different data visualisations and quite similar performance. This is because different numerical representations can differ in the relative location of clusters (in the two-dimensional data space). Therefore, in different applications it can result in a different result. Here contigs of different species form visually separate clusters with a very limited overlap with the clusters of other species. Average run time in seconds is also provided in Table III. The run time includes loading the data, numerically representing the data, MLBP feature extraction, and dimension reduction using BH-tSNE.

TABLE II
PRECISION, RECALL, F1 SCORE (%), AND THE NUMBER OF CLUSTERS FOR VARIOUS NUCLEOTIDE MAPPINGS FOR SIMULATED METAGENOMIC DATA.

Nucleotide Mapping	Atomic	EIIP	Paired	Real	Integer
Precision	93.49	93.57	74.23	92.57	93.43
Recall	93.62	93.60	79.91	91.27	93.77
F1 score	93.56	93.62	76.96	91.91	93.60
Number of clusters	11	11	12	11	10

TABLE III
NUMBER OF CONTIGS, THEIR TOTAL LENGTH, AND RUN TIME(S) FOR THE SIMULATED METAGENOMIC DATA.

Number of Contigs	Total length	Run time
2184	33138556	126.51

B. Effect of MLBP Window Length

The number of features is related to the window length of the MLBP method and can effect final performance. Consequently, to check its effect, we considered various lengths of MLBP windows (Table IV and supplementary Figure 2). Here, run time only includes the time to numerically represent the data and MLBP feature selection.

Considering small window lengths, the feature space cannot describe the underlying structure of the metagenomic dataset, while a large feature vector increases the time complexity (Table IV). Hence, window size should be sufficiently large to maintain the distinctness of the signal (information regarding texture changes across various contigs).

C. Effect of RSVD

The computational complexity of our method increases as the dimensions of the features space increase. Therefore, we considered how keeping different numbers of eigen factors can effect the performance and run time of our method (Figure 5). Here, EIIP is considered for nucleotide mapping and $p \leq 6$ for

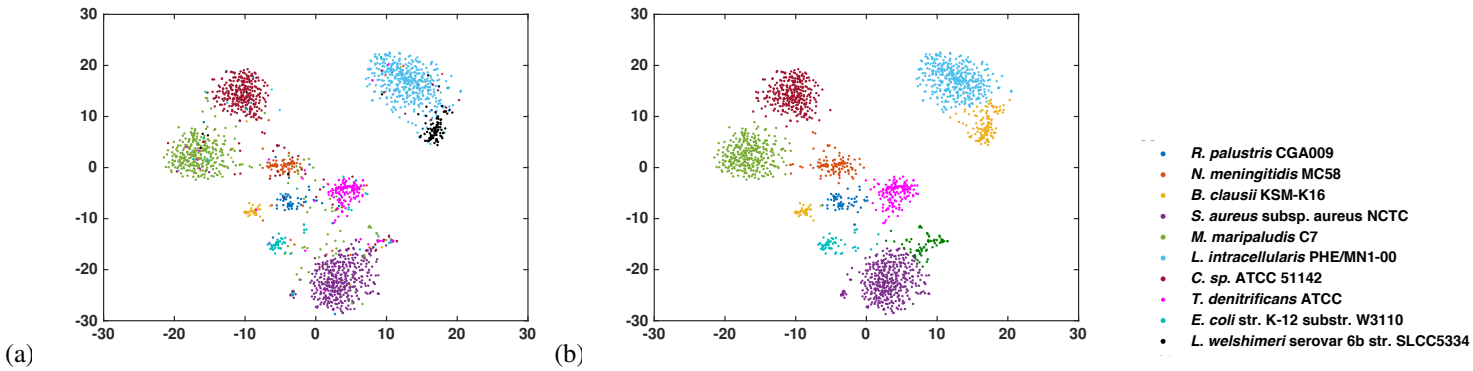

Fig. 4. Visualisation of the simulated metagenomic community using EIIP nucleotide mapping, MLBP to extract features, RSVD feature reduction, BH-tSNE for two-dimensional representation and cluster identification using DBSCAN comparing (a) manually annotated clusters (see species names in key) to (b) the DBSCAN defined clusters.

TABLE IV
PRECISION, RECALL, F1 SCORE (%), THE NUMBER OF CLUSTERS, AND THE RUN TIME(S) FOR MLBP OF VARIOUS WINDOW LENGTH ($p \leq P$).

Window Length	2	4	6	8
Precision	81.80	92.89	93.57	92.38
Recall	82.27	93.60	93.60	92.72
F1 score	82.04	93.62	93.62	92.55
Number of clusters	11	11	11	10
Run time	47.34	56.17	70.22	86.19

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 V). 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.

D. Real Data: Infant Human Gut

A relatively low-complexity infant human gut dataset was analysed to test the performance of the Multi-resolution Genomic Binary Patterns method with real data. Here, EIIP 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 algorithm binned the data into 19 clusters with precision and recall of 88.34 and 97.22. BH-tSNE representation of the data demonstrates the genomic contigs of the same or very similar contigs are binned together (Figure 6). 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 6). 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).

To 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 (Figure 7). 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 (Figure 7). For example, the clusters of *Propionibacterium* and *Peptoniphilus* species have similar abundance patterns (Clusters 9-10). Similar results were also found in [11] 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

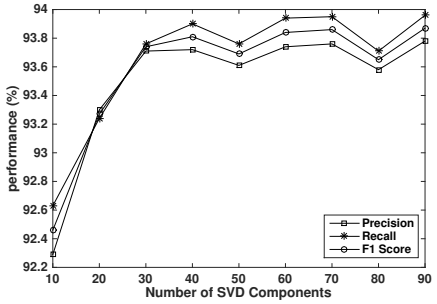

Fig. 5. Precision, recall, and F1 score (%) by keeping different numbers of RSVD components. EIIP representation and $p \leq 6$ have been considered to analyse the simulated metagenomic dataset.

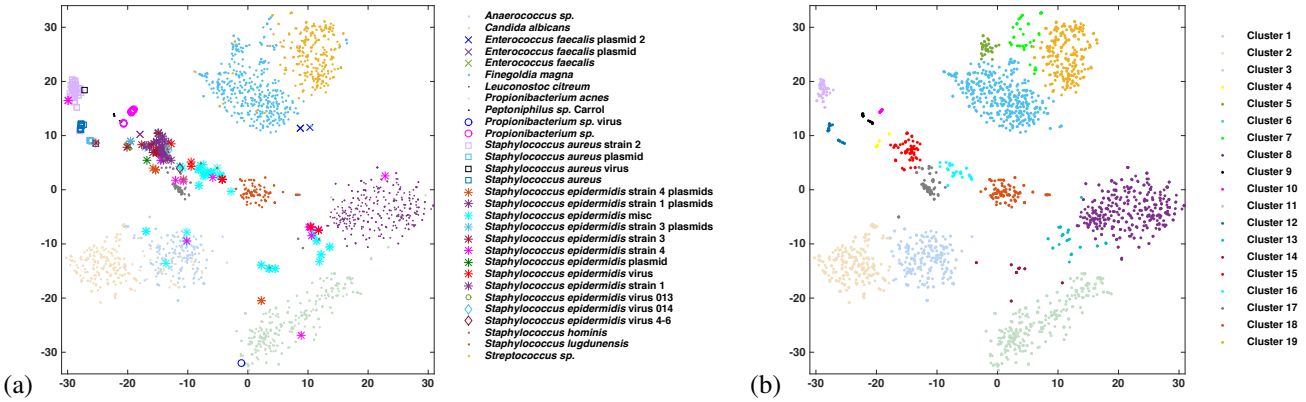

Fig. 6. Visualisation of the infant gut metagenomic community using EIIP nucleotide mapping, MLBP to extract features, RSVD feature reduction, BH-tSNE for two-dimensional representation and cluster identification using DBSCAN comparing (a) manually annotated clusters (see bacteria species, virus or plasmid names in key) to (b) the DBSCAN defined clusters 1 to 19.

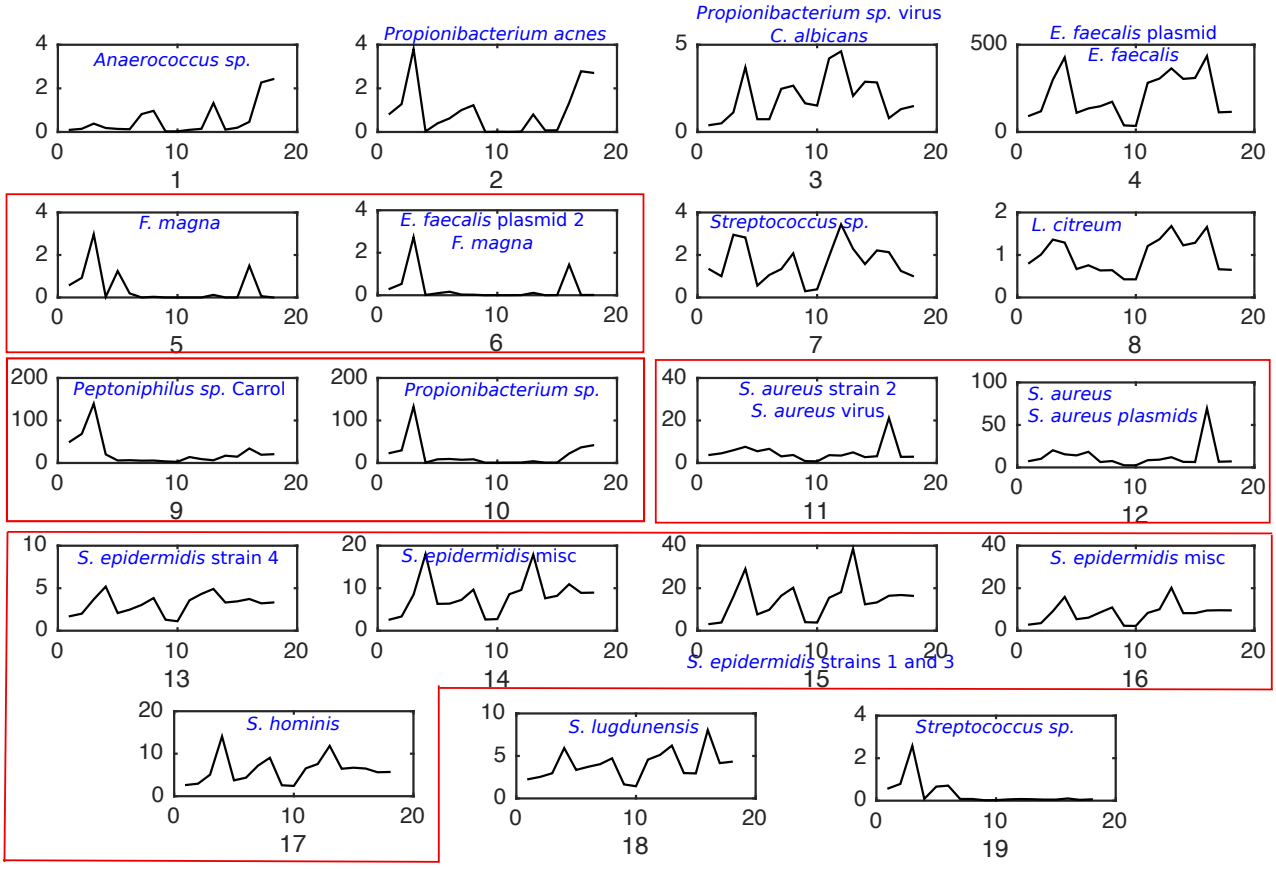

Fig. 7. Longitudinal abundance patterns of the 19 identified clusters, see Figure 6. The associated species or groups of species are indicated for each cluster. The x-axis corresponds to the longitudinal sampling over nine days [11]. The y-axis corresponds to normalised read coverage. The red box indicate the correlated clusters across longitudinal samples.

have a better separation of the related strains and viruses.

Our results compared favourably with CONCOCT [10], MetaBAT [12], and MaxBin2 [33], [34]. CONCOCT bins the data by employing sequence composition and across-sample coverage. The method has been compared with a range of methods including MetaWatt [35], SCIMM [36], and CompostBin [37] to show its advantage over composition based techniques. However, the shortcoming of the COCOCT method is having many small bins as the output (a large group of contigs may break to many clusters). MetaBAT bins the metagenomic data using probabilistic distances of genome abundance with sequence composition. It is an efficient method for analysing complex metagenomic data. However, both COCOCT and MetaBAT need large numbers of samples to perform well. 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. However, MetaBAT and MaxBin2 produce many unclassified contigs. Consequently, they have higher precision but lower recalls. Our proposed MLBP method introduces a new feature space and compares well to these three methods. The results show better performance on this dataset with small sample size (11 samples) in comparison with the other techniques (Table VI).

Finally, we checked the run time of our method. It takes about three minutes (182.71 s) to analyse this dataset (the number of contigs is 2293 and total length of them is 27594702). Although the code is relatively fast, it could be further optimised in terms of both time and memory.

IV. Conclusion

Here we have demonstrated that image processing techniques can be applied to 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 describe the genomic signature changes followed by a dimensionality reduction step 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. As illustrated in Section 3, 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.

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

Number of RSVD Components	10	20	30	40	50
RSVD run time	0.34	0.57	0.90	1.19	1.62
BH-tSNE run time	50.22	49.51	49.98	50.37	50.23
Number of RSVD Components	60	70	80	90	100
RSVD run time	1.95	2.14	2.86	3.40	3.97
BH-tSNE run time	50.80	51.19	51.25	51.47	53.00

TABLE VI
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	79.92	97.58	87.90	32
MetaBAT	85.46	93.66	89.40	10
MaxBin2	82.94	93.75	88.00	10

Acknowledgements

We would like to thank Bede Constantinides for help with metagenomics data analysis and Santosh Tirunagari for helpfull comments.

Funding

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.

References

[1] B. E. Blaisdell, "A measure of the similarity of sets of sequences not requiring sequence alignment," *Proceedings of the National Academy of Sciences*, vol. 83, no. 14, pp. 5155–5159, 1986.

[2] B. E. Blaisdell, "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*, vol. 21, no. 3, pp. 278–288, 1985.

[3] S. S. Mande, M. H. Mohammed, and T. S. Ghosh, "Classification of metagenomic sequences: methods and challenges," *Briefings in bioinformatics*, p. bbs054, 2012.

[4] M. Pietikäinen and T. Ojala, "Texture analysis in industrial applications," in *Image Technology*. Springer, 1996, pp. 337–359.

[5] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. Ho, "Detection of face spoofing using visual dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 762–777, 2015.

[6] F. Alegre, A. Amehraye, and N. Evans, "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*. IEEE, 2013, pp. 1–8.

[7] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[8] J. V. Lorenzo-Ginori, A. Rodríguez-Fuentes, R. G. Abalo, and R. S. Rodriguez, "Digital signal processing in the analysis of genomic sequences," *Current Bioinformatics*, vol. 4, no. 1, pp. 28–40, 2009.

[9] A. Tapinos, B. Constantinides, D. B. Kelland, and D. L. Robertson, "Alignment by the numbers: sequence assembly using reduced dimensionality numerical representations," *bioRxiv*, p. 011940, 2014.

[10] J. Alneberg, B. S. Bjarnason, I. de Bruijn, M. Schirmer, J. Quick, U. Z. Ijaz, L. Lahti, N. J. Loman, A. F. Andersson, and C. Quince, "Binning metagenomic contigs by coverage and composition," *Nature methods*, vol. 11, no. 11, pp. 1144–1146, 2014.

[11] I. Sharon, M. J. Morowitz, B. C. Thomas, E. K. Costello, D. A. Relman, and J. F. Banfield, "Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization," *Genome research*, vol. 23, no. 1, pp. 111–120, 2013.

[12] D. D. Kang, J. Froula, R. Egan, and Z. Wang, "MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities," *PeerJ*, vol. 3, p. e1165, 2015.

[13] C. C. Laczny, T. Sternal, V. Plugaru, P. Gawron, A. Atashpendar, H. H. Margossian, S. Coronado, L. Van der Maaten, N. Vlassis, and P. Wilmes, "VizBin-an application for reference-independent visualization and human-augmented binning of metagenomic data," *Microbiome*, vol. 3, no. 1, p. 1, 2015.

[14] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[15] M.-S. Paukkeri, I. Kivimäki, S. Tirunagari, E. Oja, and T. Honkela, "Effect of dimensionality reduction on different distance measures in document clustering," in *International Conference on Neural Information Processing*. Springer, 2011, pp. 167–176.

[16] D. Deng and N. Kasabov, "ESOM: An algorithm to evolve self-organizing maps from on-line data streams," in *Proc. of IJCNN*, vol. 6, 2000, pp. 3–8.

[17] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms." *Journal of machine learning research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[18] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions," 2009.

[19] B. Cleary, I. L. Brito, K. Huang, D. Gevers, T. Shea, S. Young, and E. J. Alm, "Detection of low-abundance bacterial strains in metagenomic datasets by eigengenome partitioning," *Nature biotechnology*, vol. 33, no. 10, pp. 1053–1060, 2015.

[20] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with bowtie 2," *Nature methods*, vol. 9, no. 4, pp. 357–359, 2012.

[21] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[22] R. F. Voss, "Evolution of long-range fractal correlations and 1/f noise in DNA base sequences," *Physical review letters*, vol. 68, no. 25, p. 3805, 1992.

[23] R. Ranawana and V. Palade, "A neural network based multi-classifier system for gene identification in DNA sequences," *Neural Computing & Applications*, vol. 14, no. 2, pp. 122–131, 2005.

[24] B. Demeler and G. Zhou, "Neural network optimization for E. coli promoter prediction," *Nucleic acids research*, vol. 19, no. 7, pp. 1593–1599, 1991.

[25] A. S. Nair and S. P. Sreenadhan, "A coding measure scheme employing electron-ion interaction pseudopotential (EIIP)," *Bioinformation*, vol. 1, no. 6, pp. 197–202, 2006.

[26] P. Bernaola-Galván, P. Carpena, R. Román-Roldán, and J. Oliver, "Study of statistical correlations in DNA sequences," *Gene*, vol. 300, no. 1, pp. 105–115, 2002.

[27] T. Holden, R. Subramaniam, R. Sullivan, E. Cheung, C. Schneider, G. Tremberger Jr, A. Flamholz, D. Lieberman, and T. Cheung, "ATCG nucleotide fluctuation of Deinococcus radiodurans radiation genes," in *Optical Engineering+ Applications*. International Society for Optics and Photonics, 2007, pp. 669 417–669 417.

[28] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[29] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin *et al.*, "The sequence alignment/map format and samtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.

[30] H. Li, "A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data," *Bioinformatics*, vol. 27, no. 21, pp. 2987–2993, 2011.

[31] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[32] S. Boisvert, F. Raymond, É. Godzaridis, F. Laviolette, and J. Corbeil, "Ray Meta: scalable de novo metagenome assembly and profiling," *Genome biology*, vol. 13, no. 12, p. 1, 2012.

[33] Y.-W. Wu, Y.-H. Tang, S. G. Tringe, B. A. Simmons, and S. W. Singer, "MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm," *Microbiome*, vol. 2, no. 1, p. 1, 2014.

[34] Y.-W. Wu, B. A. Simmons, and S. W. Singer, "Maxbin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets," *Bioinformatics*, p. btv638, 2015.

[35] S. Chatterji, I. Yamazaki, Z. Bai, and J. A. Eisen, "CompostBin: A DNA composition-based algorithm for binning environmental shotgun reads," in *Annual International Conference on Research in Computational Molecular Biology*. Springer, 2008, pp. 17–28.

[36] D. R. Kelley and S. L. Salzberg, "Clustering metagenomic sequences with interpolated markov models," *BMC bioinformatics*, vol. 11, no. 1, p. 1, 2010.

[37] A. Kislyuk, S. Bhatnagar, J. Dushoff, and J. S. Weitz, "Unsupervised statistical clustering of environmental shotgun sequences," *BMC bioinformatics*, vol. 10, no. 1, p. 1, 2009.