Marginalised Stack Denoising Autoencoders for Metagenomic Data Binning

Samaneh Kouchaki*‡,Santosh Tirunagari†, Avraam Tapinos*, David L Robertson*§
*Evolution and Genomic Sciences, School of Biological Sciences, The University of Manchester, UK
†Department of Computer Science, University of Surrey, UK
‡Department of Engineering Science, University of Oxford, UK, Email: samaneh.kouchaki@eng.ox.ac.uk
§MRC-University of Glasgow Centre for Virus Research, Glasgow, UK

***Abstract*—Shotgun sequencing has facilitated the analysis of complex microbial communities. Recently we have shown how local binary patterns (LBP) from image processing can be used to analyse the sequenced samples. LBP codes represent the data in a sparse high dimensional space. To improve the performance of our pipeline, marginalised stacked autoencoders are used here to learn frequent LBP codes and map the high dimensional space to a lower dimension dense space. We demonstrate its performance using both low and high complexity simulated metagenomic data and compare the performance of our method with several existing techniques including principal component analysis (PCA) in the dimension reduction step and k-mer frequency in feature extraction step.**

I. Introduction

An enormous volume of biological data is generated using next-generation sequencing technology. To understand the underlying genetic information, sequence analysis is therefore an important step. The sequenced data can include a community of viruses and bacteria and reads and be from the same or different genomes that makes reconstructing individual genomes problematic. Moreover, we are presented with fragmented assemblies due to insufficient coverage, sequencing errors, sequence repetition, and genetic diversity. Consequently, alignment-free techniques [1], [2] have been introduced as an alternative way to analyse metagenomic data [3] by incorporating species-specific genomic *signatures* extracted by calculating the normalised frequency of k-mers of a specific size, e.g., commonly $k = 4$. This frequency is obtained by counting the occurrences of each k-mer combination and represents a feature vector in high-dimensional space.

Recently, we have introduced an alternative feature space, local binary patterns (LBP) and its extesnion multi-resolution LBP from image processing, to extract the local changes in a sequence [4], [5]. LBP originally is an image processing feature descriptor representing local texture changes [6]. Its one-dimensional extension also found application to other signal processing areas including speech processing [7]. The problem of automatically grouping reconstructed genomic fragments into species-level groups ('binning') is considered. 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.

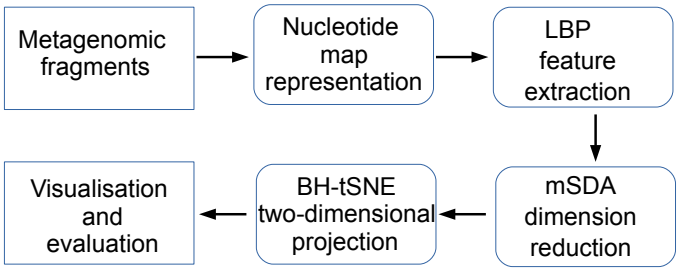

Fig. 1: Schematic overview of the proposed visualisation and binning of the metagenomic contigs.

Using LBP each sequence is represented as a high dimensional sparse feature vector. To improve the results, marginalised stacked denoising autoencoders (mSDA) [8], [9] have been used instead of common dimension reduction techniques such as principal component analysis (PCA) which captures low dimensional (dense) features. This technique has been introduced to reconstruct frequent words from infrequent ones with application to text mining [8]. It is a fast and an unsupervised method that can reliably capture important patterns in the data. After extracting important patterns, Barnes-Hut t-distributed stochastic neighbour embedding (BH-tSNE) [10] is used for visualisation and binning step.

We then compared our results with a number of metagenomic techniques in terms of precision, recall, and F1 score. Our results show the application of the proposed method to the analysis of contigs from a single sample metagenomic study.

II. Methodology

In this section, our pipeline (Fig. 1) is explained in more detail. We numerically represent the genomic fragments using a nucleotide mapping. After that LBP is used to extract features from these numerical representations. mSDA is then used to reduce the dimensions of the LBP feature vectors by capturing the frequent dense patterns. BH-tSNE is then used to map data onto a two-dimensional space for visualisation and binning. For quantitatively evaluating the visualisation performance, we cluster the BH-tSNE projected data using DBSCAN algorithm and calculate precision, recall, and F1 score between the DBSCAN assigned labels and the original labels.

TABLE I: EIIP, atomic, and paired nucleotide representations.

Letter	EIIP	Atomic	Paired
A	0.1260	70	0
C	0.1340	58	1
G	0.0806	66	1
T	0.1335	78	0

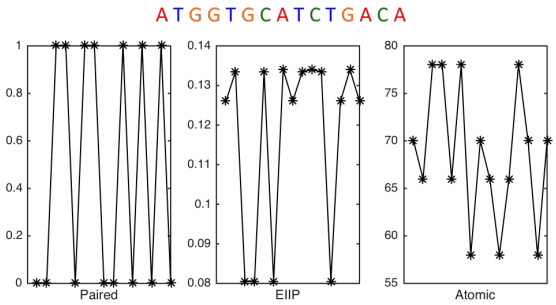

Fig. 2: A nucleotide sequence (top) and its EIIP, atomic, and paired representations. Depending on the representation, each letters A, C, G, and T of nucleotide sequence is mapped to a specific value.

A. The Nucleotide Mapping

To represent the data in terms of a LBP feature vector, the data should be represented numerically. There are two main numerical representation groups for the genomic contigs: (1) assigning an arbitrary value to each letter A, C, G, and T of the nucleotide sequence including two and four bit binary representations [11], [12] and Voss representation [13]. (2) Assigning a number based on certain biochemical or biophysical properties of the DNA molecules including paired nucleotide representations [14], EIIP [15], and atomic representations [16]. Table I shows the value assigned to each nucleotide in each of the representations. Fig. 2 shows an example of mapping a nucleotide sequence to three numerical vectors.

B. Local Binary Patterns

LBP has found popularity not only in the field of image processing but also in signal processing [17]. Using traditional two-dimensional LBP, each data window is transferred to a fixed length binary number. LBP codes illustrate the data pattern and the corresponding distribution indicates how often each pattern appears. The LBP distribution of genomic fragments is considered as the species specific genomic signatures and our feature vectors in this work.

LBP examines the neighbouring points of each data point and assigns a binary code to it. By considering $x(t)$ as the numerical representation of the tth position of a genomic segment, LBP is defined as

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

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

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

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. Finally, by considering all the obtained codes, the distribution of the LBP codes can be defined as

$$h_k = \sum_{p/2 \leq i \leq N-p/2} \delta(\text{LBP}_p(x(i),k)), \tag{3}$$

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. Here, one-dimensional LBP is considered as a feature space to compare genomic pattern changes.

C. Marginalised *Stacked Denoising Autoencoders (mSDA)*

mSDA, developed by Chen *et al.* [8], are not only computationally less expensive when compared to SVD or stacked denoising autoencoders (SDAs) but also scalable to high-dimensional features. Since we are working with genomic data, we naturally expect some noise, therefore denoising autoencoders are trained to reconstruct clean data from the noise corrupted ones. We therefore use mSDA in this study for improving the LBP feature representation through extraction of nonlinear features. In mSDA, multiple mappings are learnt to reconstruct noisy features from the most frequent features and hence it can be used as a dimensionality reduction technique.

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$ represent the numerical representation of n contigs. Each contig $\mathbf{x}_j$ is then represented using LBP distribution as $\mathbf{h}_j = \{w_{j1}, \cdots, w_{j2^p}\}$ (eq. 3). Assume that the first $n_l \ll n$ contigs have corresponding labels $\{y_1, \cdots, y_{n_l}\} \in \mathcal{Y}$. Let $\mathbf{T} = \{\mathbf{w}_{t_1}, \cdots, \mathbf{w}_{t_r}\}$ shows a strict subset of $\mathbf{H}$ (LBP distribution of all contigs) with size r and $r \ll 2^p$ referred to *prototype* bins. The algorithm aims to 'convert' each bin of $\mathbf{H}$ into one or more of these prototype bins by learning a mapping $\mathbf{W} : \mathcal{R}^{2^p} \rightarrow \mathcal{R}^r$. Therefore, the LBP distribution of feature length 2^p is transformed into a combination of prototype bins of length r.

Training of the mapping matrix $\mathbf{W}$ is based on one important point: if a prototype bin already exists in some input $\mathbf{h}$, $\mathbf{W}$ should be able to predict it from the remaining bins in $\mathbf{h}$. Consequently, it artificially creates a supervised dataset from the unlabelled data by setting each bin in $\mathbf{h}$ with some probability $(1-\varphi)$. A number of so called 'corrupted' LBP bins as $\hat{\mathbf{h}}^1, \ldots, \hat{\mathbf{h}}^m$ are then formed by performing this procedure m times.

For each input $\mathbf{h}_i$, a sub-vector is created $\bar{\mathbf{h}}_i = [x_{t_1}, \cdots, x_{t_r}]^\top \in \mathcal{R}^r$ which only contains the prototype bins. A mapping $\mathbf{W} \in \mathcal{R}^{r \times 2^p}$ is then learned to reconstruct the prototype feature bins from the corrupted ones $\hat{\mathbf{h}}_i$, by minimising the squared reconstruction error,

$$\frac{1}{2nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\|\bar{\mathbf{h}}_i - \mathbf{W}\hat{\mathbf{h}}_i^j\|^2. \quad (4)$$

A constant feature is added $\hat{\mathbf{h}}_i = [\hat{\mathbf{h}}_i; 1]$ that is not corrupted. Moreover, an appropriate bias is incorporated within the mapping $\mathbf{W} = [\mathbf{W}, \mathbf{b}]$ that reconstructs the average occurrence of the prototype features. Hence, the design matrix is given as:

$$\overline{\mathbf{H}} = [\underbrace{\bar{\mathbf{h}}_1, \cdots, \bar{\mathbf{h}}_1}_{m}, \cdots, \underbrace{\bar{\mathbf{h}}_n, \cdots, \bar{\mathbf{h}}_n}_{m}] \in \mathcal{R}^{r\times nm}$$

as the m copies of the prototype bins. Similarly, m corruptions of the original inputs denoted as $\widehat{\overline{\mathbf{H}}} = [\underbrace{\hat{\mathbf{h}}_1^1, \cdots, \hat{\mathbf{h}}_1^m}_{m}, \cdots, \underbrace{\hat{\mathbf{h}}_n^1, \cdots, \hat{\mathbf{h}}_n^m}_{m}] \in \mathcal{R}^{2^p\times nm}$ that reduce the loss function in eq. (4) to:

$$\frac{1}{2nm}\|\overline{\mathbf{H}} - \mathbf{W}\widehat{\mathbf{H}}\|_F^2, \quad (5)$$

where $\|\cdot\|_F^2$ denotes the squared Frobenius norm. The solution to (eq. 5) can be obtained under closed-form as the solution to the well-known ordinary least square.

$$\mathbf{W} = \mathbf{R}\mathbf{Q}^{-1} \text{ with } \mathbf{Q} = \widehat{\mathbf{H}}\widehat{\mathbf{H}}^\top \text{ and } \mathbf{R} = \overline{\mathbf{H}}\widehat{\mathbf{H}}^\top. \quad (6)$$

Ideally, $m \to \infty$ so that by considering the weak law of large numbers, $\mathbf{R}$ and $\mathbf{Q}$ converge to their expectations and (eq. 6) becomes:

$$\mathbf{W} = E[\mathbf{R}]E[\mathbf{Q}]^{-1}, \quad (7)$$

with the expectations of $\mathbf{R}$ and $\mathbf{Q}$ defined as

$$E[\mathbf{Q}] = \sum_{i=1}^{n} E[\hat{\mathbf{h}}_i\hat{\mathbf{h}}_i^\top],\ E[\mathbf{R}] = \sum_{i=1}^{n} E[\bar{\mathbf{h}}_i\hat{\mathbf{h}}_i^\top]. \quad (8)$$

The expectations in (eq. 8) can be represented in closed-form because of the uniform corruption. The output of the linear mapping $\mathbf{W} : \mathcal{R}^d \to \mathcal{R}^r$ approximates the expected value of a prototype bin. $\tanh()$ squashing-function is applied to the output that has the effect of amplifying or dampening the feature values of the reconstructed prototype bins.

$$\mathbf{z} = \tanh(\mathbf{W}\mathbf{h}), \quad (9)$$

Algorithm 1 mSDA

function [$\mathbf{W}^s$, $\mathbf{F}^s$] = mSDA($\mathbf{H}$, r , l); l is the number of layers.
[d,n] = size($\mathbf{H}$);
$\mathbf{W}^s$ = zeros(d , d+1 , l);
$\mathbf{F}^s$ = zeros(d , n , l+1);
$\mathbf{F}^s$(: , : , 1) = $\mathbf{H}$;
for t = 1 : l **do**
    [$\mathbf{W}^s$(: , : , t), $\mathbf{F}^s$(: , : , t+1)] = mDA($\mathbf{F}^s$(: , : , t) , r);
end for

Algorithm 2 mDA

function [$\mathbf{W}$, $\mathbf{F}$] = mDA($\mathbf{H}$, r);
$\mathbf{H}$ = [$\mathbf{H}$; ones(1,size($\mathbf{H}$, 2));
d = size($\mathbf{H}$, 1);
$\mathbf{q}$ = [ones(d - 1 , 1)(1 - φ) ; 1];
$\mathbf{C}$ = $\mathbf{H}\mathbf{H}^\top$;
$\mathbf{Q}$ = $\mathbf{C}$.($\mathbf{q}\mathbf{q}^\top$);
$\mathbf{Q}$(1 : d + 1 : end) = $\mathbf{q}$.diag($\mathbf{C}$);
$\mathbf{R}$=$\mathbf{C}$.repmat($\mathbf{q}^\top$, d , 1);
$\mathbf{W}$ = $\mathbf{R}$(1 : end - 1 , :)/($\mathbf{Q}$ + (1e-5)eye(d));
$\mathbf{F}$=tanh($\mathbf{W}\mathbf{H}$);

For an input $\mathbf{h}_i$, the r most frequent feature bins are denoted as $\mathbf{z}_i \in \mathcal{R}^r$. Later reconstruction is performed with S random non-overlapping subsets of input features $\tilde{\mathbf{h}}_i = [\mathbf{h}_i^{1^\top}, \cdots, \mathbf{h}_i^{S^\top}]^\top$. For each one of these sub-spaces independent mapping $\mathbf{W}^s$ is learnt which minimises

$$\mathcal{L}_s(\mathbf{W}^s) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{s=1}^{S}\left\|\mathbf{z}_i - \mathbf{W}^s\tilde{\mathbf{h}}_i^s\right\|^2. \quad (10)$$

Finally, mSDA output is the average of all reconstructions:

$$\mathbf{f}^1 = tanh\left(\frac{1}{S}\sum_{s=1}^{S}\mathbf{W}^s\mathbf{h}^s\right) \quad (11)$$

This explains one layer dimension reconstruction of $r << 2^p$. Consequently, multiple layers can be stacked on top of the first layer as described in Algorithm 1 and Algorithm 2.

D. Barnes-Hut t-Distributed Stochastic Neighbour Embedding

BH-tSNE is used many areas as a nonlinear technique for high dimensional data visualisation [10]. It is based on the divergence minimisation of input objects distributions and the corresponding low-dimensional data points. As a result, it can preserve the original local data structure in the final lower dimension.

Normalised Gaussian kernel has been considered as an ordinary similarity measure but it scales quadratically to the number of data points. The main objective function also has been approximated by defining the similarity function based on a number of neighbouring points [10]. In addition, a vantage-point tree is employed for decreasing search complexity. BH-tSNE is then a more efficient ($O(N\log N)$) data reduction approach and is used in this paper for the second stage of dimension reduction, two-dimensional data visualisation and clustering.

E. DB-SCAN

DBSCAN [18] is a popular density-based clustering algorithm with the aim of discovering clusters from the approximate density distribution of corresponding data points. DBSCAN takes two parameters: Epsilon, ε, and the minimum amount of elements necessary to produce a cluster, minPts. The initialisation point is a random point which has not been visited previously. The ε-neighbourhood of this point

is then retrieved and if it consists of an acceptable number of elements, a cluster is formed, otherwise the element is considered as noise. This element can subsequently be located in a properly size ε-environment of some other elements and therefore be perceived as part of a cluster. In case an element appears to be a dense point of a cluster, its ε-neighbourhood is likewise a part of that specific cluster. Thus, all elements which are discovered inside the ε-neighbourhood are included, together with their own ε-neighbourhood when they are also dense. This is carried out up to the point where the density-connected cluster is formed entirely. Subsequently, an unvisited element is retrieved and then processed, resulting in the formation of a different cluster or noise.

F. Datasets

To validate the effectiveness of our methodology we consider two simulated datasets. Simulated Illumina sequences for 10 and 100 genomes were downloaded from http://www.bork.embl.de/~mende/simulated_data/. The data were assembled by Ray Meta [19] into contigs (k = 31).

G. Performance Evaluation

In order to check the performance of our method, DBSCAN [18] 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 clusters "purity". Assuming there are g genomes in the dataset that are binned to a clusters, the precision, recall, and F1 score can be calculated as

$$\begin{aligned}
\text{Precision} &= \frac{\sum_{i=1}^{a} \max_{j} s_{ij}}{\sum_{i=1}^{a}\sum_{j=1}^{g} s_{ij}} \\
\text{Recall} &= \frac{\sum_{j=1}^{g} \max_{i} s_{ij}}{\sum_{i=1}^{a}\sum_{j=1}^{g} s_{ij} + \sum \text{unbinned sequences}} \\
\text{F1} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\end{aligned} \tag{12}$$

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

III. Experimental Results and Discussion

The performance of our method is illustrated for simulated datasets using integer nucleotide representations for LBP length $p = 8$. Our experiments show integer nucleotide representations and $p = 8$ results in better performance. We first analysed the simulated low complexity dataset with 10 genomes to show the effectiveness of our method to visualise the data (Figure 3). Then, we analysed the effectiveness of various aspects of our binning considering: (1) a 4-mer feature vector instead of the LBP distribution to compare our feature space with a commonly used 4-mer frequency and (2) PCA as linear dimension reduction compared to mSDA. We

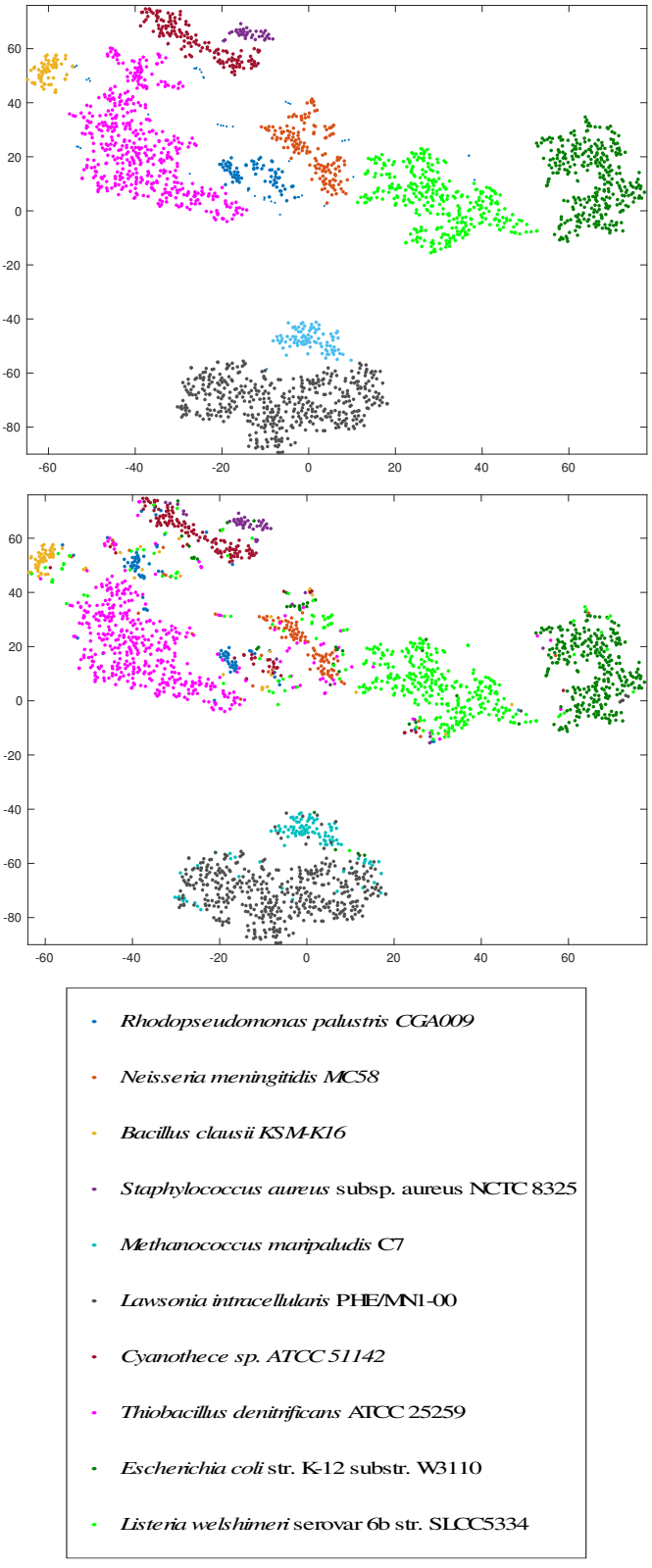

Fig. 3: Visualisation of the simulated 10 genomic community by considering LBP window length $p = 8$. Each colour represents a different species (see key) on the left side (the clusters are manually annotated) and a cluster defined by our approach on the right hand side figures.

TABLE II: Precision, recall, F1 score (%), and the number of clusters for our proposed method, 4-mer, PCA, CONCOCT, and MaxBin.

10 Genomes				
Methods	Precision	Recall	F1 score	Number of clusters
LBP	98.37	99.83	99.83	10
4-mer	96.14	70.80	81.54	13
PCA	90.41	96.33	93.27	11
CONCOCT	98.56	97.35	97.95	19
MaxBin	93.43	96.65	95.01	10
100 Genomes				
Methods	Precision	Recall	F1 score	Number of clusters
LBP	95.47	82.73	88.64	99
4-mer	95.32	69.56	80.43	98
PCA	65.60	90.67	76.13	101
CONCOCT	60.73	96.37	74.51	79
MaxBin	89.83	83.96	86.80	85

considered that the pipeline is fixed (similat to our pipeline) and only feature space or dimension reduction steps were changed. The results show that the proposed method has a more discriminative feature vector (LBP compared to 4-mer) (Table II). Furthermore, mSDA performs better than the PCA. mSDA has lower time complexity than PCA but needs more time data points to perform well. Therefore, we expected to have better results for larger datasets.

Text also compared our binning pipeline with two binning techniques: (1) CONCOCT [20] bins the data by employing sequence composition and across-sample coverage and (2) MaxBin [21], [22] that 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. CONCOCT works well for low complexity data but our results show for high complexity metagenomic data CONCOCT could not work as well as other techniques (Table II). MaxBin produced many unclassified contigs. Consequently, it has higher precision but lower recalls. For both low and high complexity genomic data, our proposed method performs better than other compared methods. It shows that the proposed pipeline can work for low and high complexity datasets.

IV. Conclusion

A metagenomic visualisation and binning approach has been implemented using LBP for feature extraction and mSDA for nonlinear feature selection. Our results on simulated genomic contigs show the underlying taxonomic structure of the metagenomic data and confirm the advantage of using image processing approaches combined with nonlinear dimension reduction techniques for metagenomic data analysis.

Acknowledgement

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 a 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] S. Kouchaki, S. Tirunagari, A. Tapinos, and D. L. Robertson, "Local binary patterns as a feature descriptor in alignment-free visualisation of metagenomic data," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2016, pp. 1–6.

[5] S. Kouchaki, A. Tapinos, and D. L. Robertson, "An image processing method for metagenomic binning: multi-resolution genomic binary patterns," *bioRxiv*, 2017. [Online]. Available: http://www.biorxiv.org/content/early/2017/04/10/096719

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

[7] 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.

[8] M. Chen, Z. Xu, F. Sha, and K. Q. Weinberger, "Marginalized denoising autoencoders for domain adaptation," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012, pp. 767–774.

[9] Z. E. Xu, M. Chen, K. Q. Weinberger, and F. Sha, "From sbow to dcot marginalized encoders for text representation," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 1879–1884.

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

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.