De novo assembly of nucleotide sequences in a compressed feature space

[1]Avraam Tapinos
[1]*Evolution and Genomic Sciences*
School of Biological Sciences
The University of Manchester
Email: avraam.tapinos@manchester.ac.uk

[1,2]David L Robertson
[2]*MRC-University of Glasgow*
Centre for Virus Research
Garscube Campus
University of Glasgow
Email:david.l.robertson@glasgow.ac.uk

***Abstract*—Sequencing technologies allow for an in-depth analysis of biological species but the size of the generated datasets introduce a number of analytical challenges. Recently, we demonstrated the application of numerical sequence representations and data transformations for the alignment of short reads to a reference genome. Here, we expand out approach for *de novo* assembly of short reads. Our results demonstrate that highly compressed data can encapsulate the signal sufficiently to accurately assemble reads to big contigs or complete genomes.**

Keywords: High-throughput sequencing, *de novo* assembly, data transformation, data compression, DFT, DWT, compressive genomics

1. Introduction

Contemporaty sequencing technologies can generate very high numbers of reads capturing the nucleotide order within genetic material [1]. This allows rapid sequencing of samples, but the resulting datasets consist of many reads that are significantly shorter than the genome of interest. These short reads are assembled to bigger continuous sections (contigs) that characterise regions of the given DNA/RNA genome. Different approaches have been proposed for handling short read data, plus numerous implementations have been developed allowing fast data analysis [2].

If the composition of the data is known a *priori*, reference based aligner mappers can be ideal candidates for data analysis. Reference base mappers are used to align reads to larger contigs or entire genomes by matching data to the regions of the known reference sequence. This approach is fast and relatively computationally inexpensive to perform [3]. However, the results are biased towards the reference sequence, and the approach can only be used if a reference sequence is known. The tools Bowtie2 [4] and BWA [5], utilise suffix array and Burrows-Wheeler Transform approaches for fast data analysis.

For cases where reference sequences are not available, or the species composition of the metagenome is not known, *de novo* assembly is more suitable. *De novo* assemblers join reads or parts of reads to extract bigger contigs from the data themselves [6]. These techniques are more computationally expensive than reference based mapping, and the results can vary considerably due to data variation, the amount of sequence repetition, coverage and genetic diversity. However, *de novo* assembly results are not biased by the reference genome and can be used for identifying unknown organisms in the data [7]. *De novo* assembler tools include SPAdes [8] and Newbler [9] which utilise *k*-mer de Bruijn graphs or overlap layout consensus (OLC) graph approaches.

Modern *de novo* assemblers generally leverage either de Bruijn graphs or read overlap graphs as part of the approach known as OLC graphs [10]. The de Bruijn graph approaches identify all the unique *k*-mers (shorter than the actual reads) within the dataset. Each unique *k*-mer is considered as an edge and is used to construct a graph by connecting pairs of edges with overlap of (*k*-1)-mers. Subsequently, graph algorithms are applied to the contigs that exist within the dataset [11].

In the OLC *de novo* approach whole reads are considered as nodes and overlaps within the reads are considered as edges. If two reads share a similarity of at least a specific overlap size then they are connected by an edge. After comparing every possible overlaps with the reads, a graph is generated and graph algorithms are used to identify the consensus contig sequence [12].

De Bruijn and OLC *de novo* approaches can be time consuming and require large amounts of computational resources for data analysis. Furthermore, in cases of data with high variation, actual similarities can be overlooked due to the the presence of mismatches

Furthermore, another challenging aspect we need to overcome during short reads analysis is the enormous size of the dataset. If a datasets size exceeds the capacity of a computers random access memory (RAM), data must be exhaustively swapped between RAM and disk storage that is orders of magnitude slower to access, forming a major analysis bottleneck [12].

De novo assembly in particular requires far greater memory than is needed to store the read information itself. Also, the pairwise data comparison stage is cumbersome, thus suffix array indexing structures such as the BWT and FM-index are widely used to reduce the data processing stage. However, these methods perform the data analysis on the original texture representations of the nucleotide sequences,

and are inadequate for datasets with high rates of variation or repetition. Such indexing structures cannot process mismatches within reads, necessitating the use of more costly alignment-based algorithms for sensitive searches. To handle high numbers of mismatches and reduce the running time of *de novo* assemblies, aggressive heuristics are employed, which in turn can compromise assembly quality.

Contemporary sequencing technologies are generating longer reads 10_s of thousands bases long. The increased length of the data will amplify existing problems with the searching techniques and it introduce new challenges for data analysis related to the curse of dimensionality. High dimensional data appear sparse in space, thus data organisation into meaningful groups becomes more challenging [13].

Comparable analytical challenges involving enormous datasets of high dimensional sequential data are encountered in other data-intensive fields such as signal, image processing, and time series analysis, where a number of effective dimensionality reduction methods have been proposed, including the use of the discrete Fourier transform (DFT) [14], the discrete wavelet transform (DWT) [15], [16], and piecewise aggregate approximation (PAA) [17], [18]. These techniques are used to transform data to an alternative feature space that allows easier identification of major feature characteristics that are not easily observed in the original data. The minor features of the data, like noise, are removed and the data analysis is performed on the reduce data approximations allowing faster more efficient data analysis that overcomes the curse of dimensionality.

We recently proposed the used of signal processing data transformation methods for aligning reads to a reference genome. The results of our proposed data transformation approach, despite data compression, was comparable to the performance of state of the art reference mappers [19].

Here we aim to evaluate the performance of data transformation and compression techniques for the *de novo* analysis of short reads. The majority of current data analytic techniques process the reads in their original space, and have to either resolve in different heuristics or use of computationally expensive matching algorithms to identify similarities between reads that contain high variation levels. To the contrary, data transformations can be used to capture the main features that exist within the data and suppress any variation/noise thus, allowing a better assemble of reads with similar characteristics.

2. MATERIALS AND METHODS

2.1. Symbolic to numerical sequence representations

Numerous methods have been proposed for mapping nucleotide sequences to numerical spaces. Some methods like the electron ion interaction potential (EIIP) [20] (Figure 1A), atomic method [21] (Figure 1B) aim to mimic the biochemical or biophysical properties of DNA molecules. However, these approaches can introduce some inter nucleotide bias that does not exist biologically. In contrary, methods like the Voss indicators [22] (Figure 1D) and the tetrahedron nucleotide mapping [23] approach provide a uniform distribution of distance between all nucleotides. Figure 1 depicts some presentation methods proposed for mapping nucleotide sequence at numerical space.

For our experiment we primarily focus on the Voss representation [22] (Figure 1D). This is a fixed numerical mapping approach, which converts a n-dimensional nucleotide sequences to a $4xn$ matrix. Each of the 4 rows in the matrix represents a nucleotide and each column represents the nucleotide in the particular position at the sequence. Equation 1 is used to generate the Voss indicator matrix where a binary value of 1 is assigned to the cell ji^{th} (where j symbolize the row value and i the column value) in indicating the existence of the j^{th} nucleotide in the i^{th} position of the sequence and a value 0 for the absence of the nucleotide.

$$V_{4i} = \begin{bmatrix} 1i \begin{cases} 1 \; if \; i \; = \; A \\ 0 \; otherwise \end{cases} \\ 2i \begin{cases} 1 \; if \; i \; = \; C \\ 0 \; otherwise \end{cases} \\ 3i \begin{cases} 1 \; if \; i \; = \; G \\ 0 \; otherwise \end{cases} \\ 4i \begin{cases} 1 \; if \; i \; = \; T \\ 0 \; otherwise \end{cases} \end{bmatrix}, \; \forall \; i \; \in \; S_n \qquad (1)$$

2.2. Data Transformation

Suitable data transformations for our analysis should be able to transform data to an alternative space without loss of useful information, have low computational overheads, facilitate rapid comparison of data, and provide lower bounding [24]. The DFT and the DWT transformation methods satisfy these requirements and are widely used for analyzing discrete signals [25], we thus use them here. These, transformation techniques are also used for identifying the major features characteristics in a dataset, and subsequently can be used to generate accurate data approximations containing only the major data features. In general the approximate data transformations have a lower dimensionality than the original data thus can be used to fit bigger datasets in memory for analysis. These methods can be used to transform/approximate nucleotide sequence numerical representations to different levels of resolution, permitting reduced dimensionality sequence analysis.

The DFT decomposes a numerically represented nucleotide sequence with *N* positions (dimensions) into a series of *N* frequency components ordered by their frequency. However, since the DFT method has high time complexity, $O(N^2)$, the fast Fourier transform (FFT) algorithm [26] with lower time complexity, $O(Nlog(N))$, is typically used instead. A prerequisite of the FFT algorithm is a signal

with length equal to an integer exponent of two, 2^n. Where sequences have a length other than 2^n, they are padded with zeros up to the next integer exponent of two prior to application of the FFT. Furthermore, the DFT decomposition of a real signal with sampling rate N (length) is conjugate symmetric [27]. Meaning that the second half of the frequencies decomposition mirrors the first half of the decomposition, thus we can safely discard the second half of the frequencies without any loss of information [28]. Furthermore, in time series data mining, a subset of the resulting Fourier frequencies are used to approximate the original sequence in a lower dimensional space [14], and the tradeoff between analytical speed and accuracy can be varied according to the number of frequencies considered [29]. Fig 1 depicts different examples of the DFT transformations of a short nucleotide sequence.

The DWT is a set of averaging and differencing functions that may be used recursively to represent sequential data at different resolutions [15], [30]. Unlike the DFT, the DWT provides time-frequency localisation, so better accommodates changes in signal frequency over time (non-stationary signals), compared with the DFT and related methods [31]. As with the FFT, a drawback of the DWT is its requirement of input with length of an integer exponent of two, (2^n). Where sequences have a length other than 2^n, artificial zero padding is again therefore added to increase the size of the signal up to the next integer exponent of two prior to application of the DWT. The corresponding DWT transformations are then truncated in order to remove the bias associated with artificial padding [32]. For example, in order to generate the DWT transformation of a time series with 500 data points to a resolution of three, i.e. 23, artificial padding must be added to increase its length to 512 (29) the next integer exponent of two. In this case, the final, eighth wavelet series should be truncated so as to avoid introducing bias. Figure 1 depicts different examples of the DWT transformations of a short nucleotide sequence.

2.3. Similarity search approaches for sequential data

Pairwise similarity/dissimilarity methods such as the Euclidian distance, the LB_keoght lower bound dynamic time warping envelope [33], longest common subsequence (LCS) [34] and alignment approaches such as the Needleman-Wunsch, Smith-Waterman and the dynamic time warping algorithms can be used to evaluate the resemblance between different data sequences. However, performing pairwise comparisons between all the data is very time consuming for big datasets ($O(n^2)$ time complexity). Indexing structures like the *KD*-tree [35], *R**-tree [36], *VP*-tree [37] and *MVP*-tree [38] are preferred approaches for fast data analysis because they allow effective and efficient data comparison. Efficient data structures require a $O(nlog(n))$ time complexity to build and the search function requires a $O(log(n))$ times.

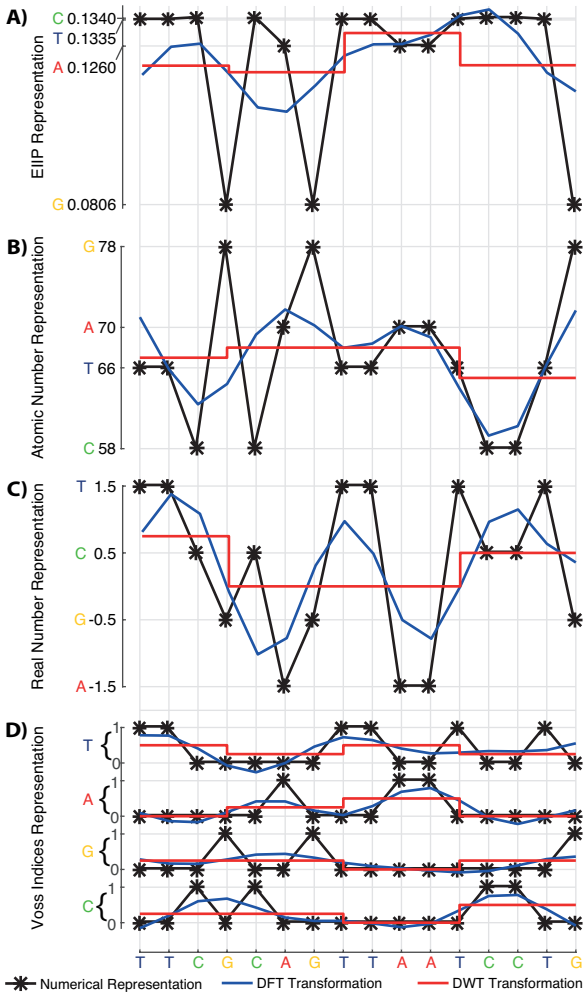

Figure 1. Examples of different numerical representations for DNA sequences and their DFT and DWT approximations. A 16-nucleotide sequence (x-axis) is represented as numerical sequences (black lines) using the EIPP (1A), the atomic number (1B), the real number (1C) and the Voss indices (1D) representation approaches. The DFT approximations of each representation method are depicted in blue and the DWT approximations of each representation method are depicted in red.

TABLE 1. METRIC PROPERTIES

i)	$d(x,y) \geq 0$	Non Negativity
ii)	$d(x,y) = 0 \Leftrightarrow x = y$	Identity
iii)	$d(x,y) = d(y,x)$	Symmetry
iv)	$d(x,z) \leq d(x,y) + d(y,z)$	Triangle inequality

Here we use a *VP*-tree structure, for data partitioning and data comparison in order to identify potential overlaps between our reads. A *VP*-tree is a metric tree, which partition data in metric space and can subsequently be used to for k nearest neighbour (k-NN) search or range search. Due to the fact that in a *VP*-tree structure data are partition based on their geometric properties, its crucial to use a similarity/dissimilarity method used that satisfies all the metric properties included in table 1. Here we are using the Euclidean distance because it is simple and fast method

that satisfies all the metric properties can be applied on the DFT, DWT and PAA data transformations/approximations

A *VP*-tree contains internal branches nodes and leaf nodes. The internal nodes contain the data point used as a vantage point *vp* and a threshold value *mu*, where as each leaf nodes only contain a data point *vp*. To initiate the construction of *VP*-tree, a data point is selected (either randomly or by applying some heuristic to find and use the furthest point in the dataset [39]) as a vantage point *vp*, and recorded the first internal branch. The pairwise distance between the rest of the data points and the selected vantage point is calculated. Following, the distance values are arranged in an ascending order and the median value is identified and recorded as a threshold value *mu* for the specific node. Subsequently, data points that have a smaller distance to the vantage point *vp* than the threshold *mu* value are directed to the left branch and the rest of the data points are directed towards the right branch. The partition step is performed recursively to each new branch until one data point is left. At that stage, the node is considered a leaf node and only contains the individual data points information.

The resulting tree can then be used either for k nearest search to identify the k most similar data points to a given query. The k-NN search is performed using the algorithm described in algorithm 1.

Algorithm 1 k-NN search algorithm for *VP*-tree

Input: Query sequence q; the desired number of k nearest neighbours; distance s indicating the distance between q and the i^{th} nearest neighbour ($i \leq k$); the set W of the k nearest neighbours obtained so far; and node n of the tree (usual the root node).
Output: Update values of W and s.

```
Initialisation :
dist:= d(q, n.vp)
if dist ≤ s then
    Update W and s.
end if
if n = branch_node then
    if dist ≤ n.mu then
        Search (q, k, s, W, n.left)
    end if
    if dist ≥ n.mu then
        Search (q, k, s, W, n.right)
    end if
end if
```

2.4. Graph Building and Traversal

For a *de novo* assemble of reads its required to identify similarities within reads and building a graph accordingly. Then the graph is traversed in order to obtain biggest contigs that characterize the data. For the de Bruijn graph approach reads are reduced to k-mers. Each k-mer is considered as a node and any k-1 overlap between different nodes is considered as an edge. The overlap information are used to build the de Bruijn graph. Eulerian paths within the graph are considered as contigs. However, the use of k-mers can results into loss of read coherence, meaning that k-mers from different reads may be co-assembled. Also, the de Bruijn approach cannot identify overlaps if high variation exists within the data.

Algorithm 2 Breadth first search traversal

Input: Empty set S to hold all visited nodes; empty set Q to hold all the nodes in queue and a root node r
Output: Update list S with all visited nodes.

```
Initialisation :
Add r to S.
Add r to Q.
while Q is not empty do
    node n is the top node in Q.
    for each node m that is adjacent to n do
        if m is not in S then
            add m to S.
            add m to Q.
        end if
    end for
    remove noden from Q.
end while
```

OLC assemblers use big overlaps within reads to construct a graph. Each read is represented as a node, and subsequently merge overlapping reads into consensus contigs [40]. OLC is relatively time and memory intensive, scaling poorly to millions of reads and beyond but can be easily implemented to tolerate long and noisy sequences. We aim to implement an OLC approach for *de novo* assembly of reads. To reduce the time and space complexity we are using the *VP*-tree structure for fast identification of reads overlaps and graph building. The use the tree can alleviate the execution time of graph building.

The generated graph can be traversed using a breath first search (BFS) algorithm [41]. The BFS can be performed using the pseudo-code in algorithm 2.

3. Results

To assess the performance of our sequence transformation approach we implemented a *de novo* reads assembler as described in algorithm 3. The main key stages of our approach are as follows: *i)* transforming nucleotide sequences into numerical sequences, *ii)* creating approximate transformations of sequences and building the *VP*-tree, *iii)* performing accelerated comparison of the sequence approximations created in the previous step in order to identify candidate alignments, *iv)* build the graph based on the information form step *iii* and traverse it using a BFS algorithm, and *v)* build the overlap graphs and extract the consensus sequences.

3.1. Read simulations

The prototype *de novo* assembler implementation of our proposed approached was assessed on assembling short

Algorithm 3 *De novo* assembly using data transformations

Input: A set nucleotide sequences *S*; a representation method *R*; a transformation method *T*; minimum overlap value *o*; and a *k* of nearest neighbours.

Output: Return a set of contigs *C* obtained from the overlapped assembled reads assembled.

Initialisation :

1) Create a representation for the each read in set *S* using representation method *R*.

2) Use a sliding window approach and obtain each *o* size subsequence from the reads' representations.

3) Transform reads subsequence representations obtained from step 2 to a lower dimensional space using *T* transformation method.

for each sequence *s* in *S* **do**

4) Use algorithm 1 to identify *k* nearest neighbours for query *s*.

5) Evaluate *k* nearest neighbours using full resolution data and discard any false positives.

6) Use sequence *s* and is neighbours as nodes in a graph and connect them with edges.

end for

7) Use algorithm 2 to traverse graph generated by step *6*

8) Generate set of contigs *C* from the graph traversal and the sequences overlaps.

reads simulations to bigger full contigs. For the experiment we used 16 simulated datasets of HIV-1 HXB2 populations (GenBank accession: K03455.1, 9719bp). Reads were simulated with a single read length of 400 nucleotides and respective coverage depths of approximately 80 reads. Reads were simulated *i)* in the absence of variation or sequencing error, *ii)* with nucleotide insertion/deletion rates of 15%, *iii)* with nucleotide substitution rates of 15%, and *iv)* with matching insertion/deletion and substitution rates of 15% (2%, 4%, 6%, 8%, and 10% overall variation) so as to simulate diverse populations.

3.2. De Novo assembly

To demonstrate the applicability of our approach to the *de novo* assembly of short reads, we implemented prototype OLC *de novo* aligner based on algorithm 3. Reads are first represented as numerical sequences using the Voss method. Every subsequence with size *o* (choosen overlap size) of each numerically represented read is identified and transformed to lower dimensional space using one of different given transformation methods. The transformations are then used to generate a *VP*-tree structure. Then the initial *o* sized subsequence transformation of each read is used a query in the *VP*-tree in order to obtained the best overlap matches. During the *VP*-tree search we ensure that no read matches its self. The obtained *k* closes matches are evaluated in using the original data in order to remove any false positive data. The remaining neighbours are used to construct the weighted graph. Subsequently, the breadth-first search (BFS) algorithm is used to traverse the graph and obtain the overlapping sequences. Finally the overlapping sequences are assembled to larger contigs.

We applied our *de novo* assembly algorithm to assemble simulated short read data from viral populations (HIV-1) with the DWT and DFT transformations accordingly. For each of the approach we tested different overlap lengths (between 100 and 300 bases long), and different transformation levels. For the DFT method, data approximations were constructed using the 8 first Fourier frequencies and for the DWT data were approximate using 8 wavelets.

The lowest common ancestor of each contiq was obtained from the *de novo* assemblies were evaluated using tictax [42] (https://github.com/bede/tictax). Tictax is a Web service application that uses One Codex database to classify reads and identify their lowest common ancestors. The results from the lowest common ancestry of each contig were used to calculate the overall accuracy rate of each individual *de novo* assembly with a equation 2. TN indicates the true negative results; TP indicates the true positive results, and C indicates the number of contigs return from each assembly. If an assembly is classified as HIV from tictax then is considered as a TP results either wise a false negative results.

$$A = \frac{\sum_{i=1}^{C} TP + \sum_{i=1}^{C} TN}{C} \tag{2}$$

Our implementation managed to generate large accurate contigs of the HIV-1 genome and in many cases all the contigs generated by a distinct assembly was correctly classified as HIV-1. As it can be seen in Figure 2, the results indicate that even at high level of variation (10% of combine insertion deletion and substitution variation) the use of our proposed data transformations *de novo* approach can generate contigs with an accuracy level of 0.97.

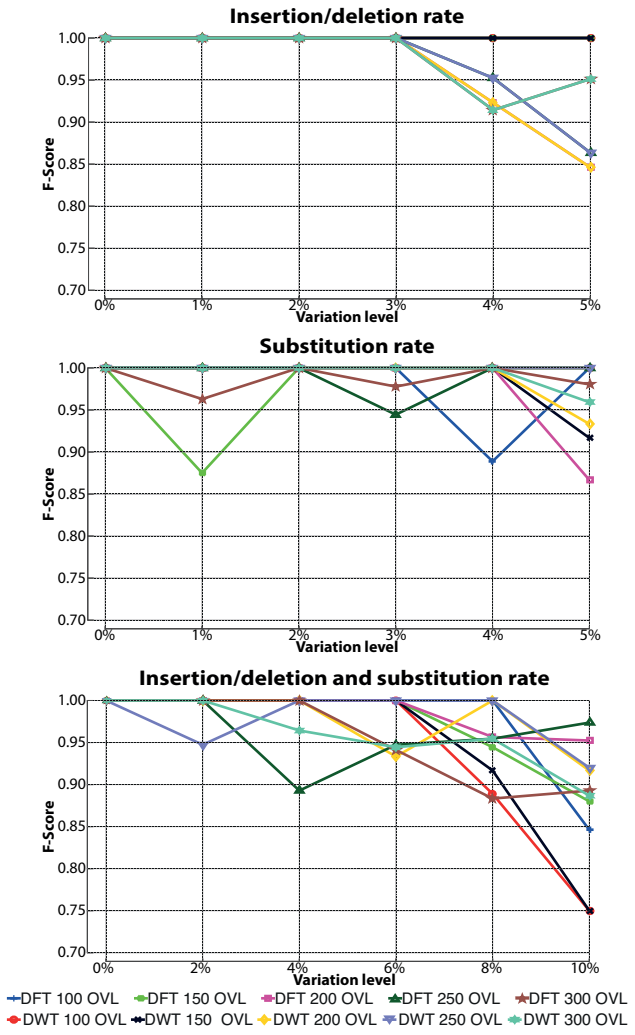

Figure 2. Accuracy of our prototype OLC assembler variants with different overlap lengths in aligning HIV-1 HXB2 simulated reads with varying levels of sequence variation. The accuracy level of the assemblies is evaluated by assessing the lowest common ancestors of the generated contigs. A show results for reads with 0-5% insertion/deletion variation, while B correspond to reads with 0-5% substitution variation. C show obtained accuracies for reads with combined, equally contributing insertion/deletion and substitution rates of 0-10%..

4. Conclusion

During the preliminary stage of genomic/metagenomic analysis short reads are assembled to larger contigs. Current state of the art technologies employ either OLC graphs or *k*-mer de Bruijn graphs. OLC graphs have high time and space complexity thus de Bruijn graphs are favoured. However, *k*-mer de Bruijn graphs efficiency can decline data contain high variation. Here we propose the use of data transformations/ approximations and signal processing techniques for the assembly of short reads. Our results indicate the use of short reads transformations and signal processing indexing techniques allow for accurate *de novo* assembly. Signal processing data transformation techniques can allowed fast and computationally inexpensive analysis of large dataset of nucleotide sequences.

5. Acknowledgement

We thank Shaun Kandathil, Bede Constandinides and Samaneh Kouchaki for valuable recommendations, data sharing and for assistance with coding. Funding: this work has been supported by the BBSRC [BB/M001121/1], and the VIROGENESIS project which receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 634650.

References

[1] S. J. Salipante, D. J. Roach, J. O. Kitzman, M. W. Snyder, B. Stackhouse, S. M. Butler-Wu, C. Lee, B. T. Cookson, and J. Shendure, "Large-scale genomic sequencing of extraintestinal pathogenic escherichia coli strains," *Genome research*, p. gr. 180190.114, 2014.

[2] R. Rose, B. Constantinides, A. Tapinos, D. L. Robertson, and M. Prosperi, "Challenges in the analysis of viral metagenomes," *Virus Evolution*, vol. 2, no. 2, p. vew022, 2016.

[3] A. M. S. Shrestha, M. C. Frith, and P. Horton, "A bioinformaticians guide to the forefront of suffix array construction algorithms," *Briefings in bioinformatics*, vol. 15, no. 2, pp. 138–154, 2014.

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

[5] H. Li and R. Durbin, "Fast and accurate short read alignment with burrowswheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[6] J. D. Kececioglu and E. W. Myers, "Combinatorial algorithms for dna sequence assembly," *Algorithmica*, vol. 13, no. 1-2, pp. 7–51, 1995.

[7] Q. Yang and X. Wu, "10 challenging problems in data mining research," *International Journal of Information Technology & Decision Making*, vol. 5, no. 04, pp. 597–604, 2006.

[8] A. Bankevich, S. Nurk, D. Antipov, A. A. Gurevich, M. Dvorkin, A. S. Kulikov, V. M. Lesin, S. I. Nikolenko, S. Pham, and A. D. Prjibelski, "Spades: a new genome assembly algorithm and its applications to single-cell sequencing," *Journal of computational biology*, vol. 19, no. 5, pp. 455–477, 2012.

[9] M. Margulies, M. Egholm, W. E. Altman, S. Attiya, J. S. Bader, L. A. Bemben, J. Berka, M. S. Braverman, Y.-J. Chen, and Z. Chen, "Genome sequencing in microfabricated high-density picolitre reactors," *Nature*, vol. 437, no. 7057, pp. 376–380, 2005.

[10] W. Zhang, J. Chen, Y. Yang, Y. Tang, J. Shang, and B. Shen, "A practical comparison of de novo genome assembly software tools for next-generation sequencing technologies," *PloS one*, vol. 6, no. 3, p. e17915, 2011.

[11] P. E. Compeau, P. A. Pevzner, and G. Tesler, "How to apply de bruijn graphs to genome assembly," *Nature biotechnology*, vol. 29, no. 11, pp. 987–991, 2011.

[12] D. C. Schwartz and M. S. Waterman, "New generations: Sequencing machines and their computational challenges," *Journal of Computer Science and Technology*, vol. 25, no. 1, pp. 3–9, 2010.

[13] N. L. Clement, L. P. Thompson, and D. P. Miranker, "Adam: augmenting existing approximate fast matching algorithms with efficient and exact range queries," *BMC bioinformatics*, vol. 15, no. Suppl 7, p. S1, 2014.

[14] R. Agrawal, C. Faloutsos, and A. Swami, *Efficient similarity search in sequence databases*. Springer, 1993.

[15] K.-P. Chan and A.-C. Fu, "Efficient time series matching by wavelets," in *Data Engineering, 1999. Proceedings., 15th International Conference on*. IEEE, Conference Proceedings, pp. 126–133.

[16] A. M. Woodward, J. J. Rowland, and D. B. Kell, "Fast automatic registration of images using the phase of a complex wavelet transform: application to proteome gels," *Analyst*, vol. 129, no. 6, pp. 542–552, 2004.

[17] P. Geurts, *Pattern extraction for time series classification*. Springer, 2001, pp. 115–127.

[18] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Locally adaptive dimensionality reduction for indexing large time series databases," *ACM SIGMOD Record*, vol. 30, no. 2, pp. 151–162, 2001.

[19] A. Tapinos, B. Constantinides, D. B. Kell, and D. L. Robertson, "Alignment by numbers: sequence assembly using compressed numerical representations," *bioRxiv*, 2015. [Online]. Available: http://biorxiv.org/biorxiv/early/2015/01/27/011940.full.pdf

[20] 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.

[21] 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, Conference Proceedings, pp. 669 417–669 417–10.

[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] B. Silverman and R. Linsker, "A measure of dna periodicity," *Journal of theoretical biology*, vol. 118, no. 3, pp. 295–300, 1986.

[24] T. Mitsa, *Temporal data mining*. CRC Press, 2010.

[25] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, *Fast subsequence matching in time-series databases*. ACM, 1994, vol. 23.

[26] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[27] W. L. Briggs, *The DFT: An Owners' Manual for the Discrete Fourier Transform*. Siam, 1995.

[28] S. Braun, *Discover signal processing: An interactive guide for engineers*. John Wiley & Sons, 2008.

[29] F. Mörchen, "Time series feature extraction for data mining using dwt and dft," 2003.

[30] A. Jensen and A. la Cour-Harbo, *Ripples in mathematics: the discrete wavelet transform*. springer, 2001.

[31] Y.-L. Wu, D. Agrawal, and A. El Abbadi, "A comparison of dft and dwt based similarity search in time-series databases," in *Proceedings of the ninth international conference on Information and knowledge management*. ACM, Conference Proceedings, pp. 488–495.

[32] D. B. Percival and A. T. Walden, *Wavelet methods for time series analysis*. Cambridge University Press, 2006, vol. 4.

[33] E. Keogh, "Exact indexing of dynamic time warping," in *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 406–417.

[34] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Data Engineering, 2002. Proceedings. 18th International Conference on*. IEEE, Conference Proceedings, pp. 673–684.

[35] B. C. Ooi, K. J. McDonell, and R. Sacks-Davis, "Spatial kd-tree: An indexing mechanism for spatial databases," in *IEEE COMPSAC*, vol. 87, Conference Proceedings, p. 85.

[36] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, *The R*-tree: an efficient and robust access method for points and rectangles*. ACM, 1990, vol. 19.

[37] A. W.-c. Fu, P. M.-s. Chan, Y.-L. Cheung, and Y. S. Moon, "Dynamic vp-tree indexing for n-nearest neighbor search given pair-wise distances," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol. 9, no. 2, pp. 154–173, 2000.

[38] T. Bozkaya and M. Ozsoyoglu, "Distance-based indexing for high-dimensional metric spaces," in *ACM SIGMOD Record*, vol. 26. ACM, Conference Proceedings, pp. 357–368.

[39] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *SODA*, vol. 93, Conference Proceedings, pp. 311–321.

[40] X. Deng, S. N. Naccache, T. Ng, S. Federman, L. Li, C. Y. Chiu, and E. L. Delwart, "An ensemble strategy that significantly improves de novo assembly of microbial genomes from metagenomic next-generation sequencing data," *Nucleic acids research*, vol. 43, no. 7, pp. e46–e46, 2015.

[41] A. Bundy and L. Wallen, *Breadth-First Search*. Springer, 1984, pp. 13–13.

[42] B. Constandinides, p. tictax: streaming sequence classification with web services, 03/04/2017 2016. [Online]. Available: https://github.com/bede/tictax