2018 IEEE International Conference on Data Mining

Predicted Edit Distance based Clustering of Gene Sequences

Sakti Pramanik and AKM Tauhidul Islam
Department of Computer Science and Engineering
Michigan State University
sakti.pramanik@gmail.com, islama@msu.edu

Shamik Sural
Department of Computer Science and Engineering
Indian Institute of Technology - Kharagpur
shamik@cse.iitkgp.ac.in

***Abstract*—Effective mining of huge amount of DNA and RNA fragments generated by next generation sequencing (NGS) technologies is facilitated by developing efficient tools to partition these sequence fragments (reads) based on their level of similarities using edit distance. However, edit distance calculation for all pairwise sequence fragments to cluster these huge data sets is a significant performance bottleneck. In this paper we propose a predicted Edit distance based clustering to significantly lower clustering time. Existing clustering methods for sequence fragments, such as, k-mer based VSEARCH and Locality Sensitive Hash based LSH-Div achieve much reduced clustering time but at the cost of significantly lower cluster quality. We show, through extensive performance analysis, clustering based on this predicted Edit distance provides more than 99% accurate clusters while providing an order of magnitude faster clustering time than actual Edit distance based clustering.**

I. Introduction

An objective of mining rapidly growing amount of DNA and RNA sequence fragments obtained from NGS technologies is to partition these huge datasets into smaller subsets based on their level of similarities. A few important applications of these smaller subsets (clusters) are to discover novel organisms and create phylogenetic relationships among them.

Numerous sequence fragments clustering methods have been proposed in the literature [1]–[5]. However, faster clustering techniques for sequence fragments, most commonly used are based on k-mer [2] and locality sensitive hashing [1]. They use mostly hamming and jaccard distance for similarity which cannot effectively capture evolutionary changes in the sequences like the ones based on Edit distance. The challenge of clustering techniques using Edit distance is to minimize the high cost of distance calculation. For example, Agglomerative hierarchical clustering techniques, frequently used by biologists [3]–[5], require all pairwise Edit distance calculations. In our approach, we use agglomerative hierarchical clustering. However, instead of Edit distance we used predicted Edit distance which requires much reduced time.

In this paper, we propose a reference sequence based space transformation technique for efficiently predicting Edit distance. Each sequence in a Database is converted into a feature vector by a set of carefully chosen reference sequences. The goal of this transformation is to significantly minimize the number of pairwise Edit distance calculations using Chebyshev distance. We use hierarchical clustering algorithm in the transformed space to create the clusters. The results show that the proposed method produces significantly better quality clusters than the existing approaches while being very fast in computation. Primary contributions of the paper are:

- We propose novel techniques for predicting Edit distances of the sequence fragments using space transformation.
- This technique guarantees accurate prediction of Edit distance one.
- Edit distance based cluster similarity thresholds can be more accurately captured by the proposed method than in k-mer or hash based techniques.
- Time requirement for the proposed clustering technique is an order of magnitude less due to much reduced cost of computing predicted Edit distance.
- Extensive performance analyses of the proposed method are provided justifying this superior performance.

The paper is organized as follows. Section II presents related works on clustering. Section III describes our proposed space transformation technique. Section IV gives the methodology for Edit distance predictions. Section V describes the adopted hierarchical clustering algorithm. Section VI reports our experimental results. Conclusion is provided in section VII.

II. Related Works

Reference sequences based space transformation has been used in many applications [6]–[9]. However, such techniques have not been used for Edit distance predictions. To the best of our knowledge, predicted Edit distance has not been used for clustering sequence fragments.

Different types of sequence similarity based clustering methods [2], [10], [11] are frequently used to analyze large sets of sequence fragments. Agglomerative hierarchical clustering algorithms of single and average linkage techniques are often used for short sequences datasets clustering. Although, these methods show satisfactory accuracy, for large datasets they can take significant amount of time. Memory complexity of these methods is in the order of $O(N^2)$ which is a serious limitation for clustering very large data sets in memory. Mothur [12] is an agglomerative hierarchical clustering algorithm. It incorporated multiple sequence alignment that reduces the computational cost of dissimilarity measurements, which results in significant speedup in clustering However, multiple sequence

978-1-5386-9159-5/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDM.2018.00160

alignments has a high time complexity ($O(N^3L)$) and space complexity ($O(N^2 + NL + L^2)$) .

Greedy heuristic algorithms such as UCLUST and VSEARCH [2], [10], [11], were proposed which cluster sequences in an online-learning paradigm. Initially, they start with a seed sequence assigned to a cluster. Then a new sequence is assigned to one of the currently existing clusters that satisfy a distance threshold with its representative sequence, forming a new cluster. These heuristics based algorithms significantly speed up the clustering at the cost of accuracy.

Locality Sensitive Hashing [1] assumes that global dissimilarity between two sequences is preserved in the randomly generated substrings of the sequences. The method uses two level hashing to reduce the number of candidate sequences. While the strategy reduces the number of pairwise comparisons, it can not calculate Edit distance of randomly generated subsets because such distance would not represent the global Edit distance between two sequences.

III. Space Transformation

Given an alphabet ω, a set of N database sequences, $\Omega = \{s_1, ..s_i., s_N\}$ and a set of n reference sequences $R = \{r_1, \dots r_j \dots, r_n\}$; a sequence in the database, s_i is mapped into an *n*-dimensional feature vector using transformation function $T(s)$:

$$T : \Omega \rightarrow Z^n$$
$$\forall s_i \in \Omega : \; T(s_i) = \mathbf{x} \quad \text{where } \mathbf{x} =< x_1, x_2, .., x_n > \text{ and } \forall x_j : x_j = dist(s_i, r_j) \tag{1}$$

$dist(s_i, R_j)$ is a distance function where it can be *Levenshtein (Edit) distance* of sequence s_i to the reference sequence R_j.

Suppose, A is a database sequence. Then, A is transformed by R into $\mathbf{x}^A =< x_1^A, x_2^A, \dots, x_j^A, \dots, x_n^A >$ where x_j^A is the j^{th} component of the feature vector $\mathbf{x}^A$.

A critical component of space transformation is the selection of an effective set of reference sequences. In the following sections, we describe some of the key properties to consider when choosing the reference sequences.

A. Length of Reference Sequences (l_{ref})

If the length of a reference sequence (l_{ref}) is smaller than the length of a database sequence, the transformation only captures the Edit distance for a portion of the database sequence plus a number of insertions (according to the definition of Edit distance). Hence, the length of reference sequences should be at least as big as the length of the sequence fragments in the database. In case the length of reference sequences are larger than the length of sequence fragments in the databases, reference sequences are trimmed at the end for consistency.

B. Distances among Reference Sequences

Feature vectors resulting from the space transformation are correlated if the reference sequences are close to each other. We want to find the proper distance between the reference sequences that gives uncorrelated feature vectors. Correlation among the reference sequences decreases monotonically with the increase of distance among each reference sequence. Reference sequences of length l_{ref} have minimum correlation when distance among each of them is l_{ref}. Therefore,

Definition 1. *We define the notation R^l as the set of reference sequences, where each sequence is of length l_{ref} and distance among them is also l_{ref}. A sequence in R^l is denoted by r^l.*

However, the size of such a set is small because as shown in the Proposition 1, the number of such sequences is the size of the alphabet, $|\omega|$.

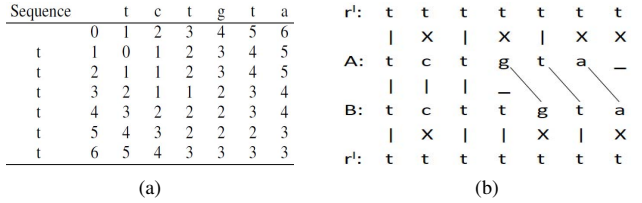
Proposition 1. *The number of reference sequences of length l_{ref} where distance among them is also l_{ref}, is exactly $|\omega|$.*

Due to space limitation, the proof is given in the complete report. Assuming $\omega = \{a, c, g, t\}$, examples of such set of reference sequences are as follows, $R^5 = \{aaaaa, ccccc, ttttt, ggggg\}, \{cacac, acaca, tgtgt, gtgtg\}$, ... However, when the letters are same, they have some unique properties which will be exploited in the following proposition.

Proposition 2. *Edit operations in Edit distance calculation between a database sequence of length l_{ref} and a reference sequence r^l consisting of same letter are guaranteed to be point mutations.*

Sequence		t	c	t	g	t	a
	0	1	2	3	4	5	6
t	1	0	1	2	3	4	5
t	2	1	1	2	3	4	5
t	3	2	1	1	2	3	4
t	4	3	2	2	2	3	4
t	5	4	3	2	2	2	3
t	6	5	4	3	3	3	3

(a)



(b)

Fig. 1: (a) Edit distance matrix between two sequences, (b) Insert operation captured through point mutation by an r^l

The proof of Proposition 2 is given in the complete report. For example, Figure 1(a) shows Edit distance calculation of two sequences. A point mutation causes the least change for each unmatched position between the two sequences.

However, as mentioned in section III-A, reference sequences can be of same length or longer than the database sequences. The above mentioned proposition considers only sequences of same length. If a reference sequence is longer than a database sequence, insert operations will occur to the database sequences for the remaining positions of the reference sequences. Figure 1(b) shows an example of this scenario.

Theorem 1. *If Edit distance between a pair of database sequences is one, difference of distances from at least one $r^l \in R^l$ consisting of same letter to each database sequence of the pair is guaranteed to be one.*

The proof of Theorem 1 is given in the complete report. Suppose, $A = tctgta$, $B = tcttga$ and $R^l = \{aaaaaa, cccccc, gggggg, tttttt\}$. Then, difference of Edit distance from at least one $r^l \in R^l$ to A and B would be one.

In addition, R^l consisting of same letter effectively capture multiple Edit distances between a pair of database sequences.

However, they cannot guarantee capturing all of them. To remedy this problem, more reference sequences of different combinations of letters are needed. Given $|w| = 4$, if we want add the fifth reference sequence, the distance among them cannot be l_{ref}. Rather, the distance among the reference sequences needed to be reduced to $\frac{3*l_{ref}}{4}$. For $l_{ref} = 300$, approximately 64 additional reference sequences can be generated for distance $\frac{3*l_{ref}}{4}$. For more reference sequences, we gradually decrease the distance requirement up to $l_{ref}/2$. This strategy ensures keeping the distance large enough to minimize the correlation as well as generating sufficient number of reference sequences with many possible combination of letters to increase the likelihood of capturing multiple Edit distances.

We experimented with 1000 pairs of sequences of length $l_{ref} = \{100, 200, 300\}$ and found that correlation among the reference sequences of distance greater than $l_{ref}/2$ are very small. Thus we produced reference sequences of distance greater than $l_{ref}/2$, getting correlation very close to zero.

The two key features of reference sequences described in this section ensure the quality of the sequences in R. Once selected, they are used to create the feature vectors in the transformed space. The values in a VD in the transformed space should closely resemble the corresponding Edit distance. In section IV, multiple heuristics are proposed to achieve this goal by using a novel Edit distance prediction technique.

IV. Edit distance prediction in transformed space

Let us consider, sequence fragments A and B in a database that are transformed by R into x^A and x^B.

Definition 2. *Vector Difference (VD): Vector representing component wise difference of two feature vectors x^A and x^B is denoted by the vector $\Delta x =< (\Delta x_j : (x_j^A \sim x_j^B)), 1 \leq j \leq n >$.*

Definition 3. *Maximum component value in a VD for a pair of feature vectors is defined by M_{vd},*

$$M_{vd} = \max\big(< (\Delta x_j : (x_j^A \sim x_j^B)), 1 \leq j \leq n >\big).$$

Definition 4. *The number of occurrences of M_{vd} in a VD for a pair of vectors is defined by F_{vd}.*

Let us assume, $x^A =< 10, 8, 12, 6, 7, 11, 9, 6, 8, 12 >$ and $x^B =< 8, 9, 15, 5, 9, 13, 6, 6, 9, 13 >$. Thus, for these two feature vectors, $VD =< 2, 1, 3, 1, 2, 2, 3, 0, 1, 1 >$, $M_{vd} = 3$ and $F_{vd} = 2$. We use M_{vd} and F_{vd} to predict the Edit distances in the transformed space. Following theorem is the basis for our proposed heuristics to predict the Edit distances.

Theorem 2. *Let Δx be the VD of two feature vector x^A and x^B. For each dimension j of Δx, $\Delta x_j \leq d$, for $1 \leq j \leq n$, where d is the Edit distance of sequence fragment A and B.*

Proof. $x_j^A = dist(A, r_j)$, $x_j^B = dist(B, r_j)$ and d is the distance between A and B. According to the triangular inequality property of Edit distance:

$$\big(x_j^A \leq \mathrm{d} + x_j^B \quad \text{and} \quad x_j^B \leq \mathrm{d} + x_j^A\big) \rightarrow \big((x_j^B \sim x_j^A) \leq \mathrm{d}\big)$$

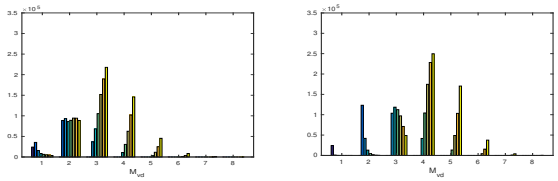
□

Corollary 1. *Given two database sequences A and B, the M_{vd} of x^A and x^B is bounded by the Edit distance of A and B.*

It should be noted that each component value in a VD greatly depends on the relative position of r_j to the pair of database sequences. Since, each r_j is quite distant from each other, adding more reference sequences increases the likelihood of higher component values in a VD. If two database sequences A and B are co-linear with a reference sequence r_j, their M_{vd} is close to the Edit distance.

Proposition 3. *Increasing number of reference sequences enhances the probability of achieving higher M_{vd}.*

According to proposition 3, M_{vd} is likely to increase with increasing number of reference sequences. Based on corollary 1, M_{vd} is bounded by the Edit distance of the corresponding two sequences. Thus with increasing number of reference sequences, M_{vd} will increase either to the Edit distance or get close to it. Therefore, with sufficiently large number of reference sequences, M_{vd} can provide a good estimation of the Edit distance in the transformed space.



(a) # Reference = 10 (b) # Reference = 50

(c) # Reference = 300 (d) # Reference = 1000

Fig. 2: Distribution of Edit distances in the transformed space grouped by M_{vd} with varying number of dimensions

A. Edit distance prediction by M_{vd}

Figure 2 shows the distribution of M_{vd} values for different Edit distances for increasing number of reference sequences (dimensions). All the sequence fragment pairs who are close to each other in the Edit space, are also close in the transformed space. For example, in Figure 2(a), 100% of the pairs with Edit distance 1 will be predicted by $M_{vd} \leq 1$. However, only 1% of the predicted pairs are actually Edit distance 1. Remaining 99% of the pairs predicted as $M_{vd} = 1$, correspond to larger Edit distances. This is because, if the Edit distance is large, in the transform space they can be close if the number of reference sequences is small. Therefore, for small number of reference sequences, a given M_{vd} corresponds to a range of Edit distances. As the number of reference sequences increases, the Edit distances become more distinguishable. For example, in Figure 2(c), for $n = 300$, if $M_{vd} = 1$, 99.9%

of them are actually Edit distance 1. Further increasing the number of dimensions will improve prediction accuracy for a given M_{vd}. For example, in Figure 2(d), for $n = 1000$ and $M_{vd} = 3$, the prediction accuracy is 95.7%. By corollary 1, if $M_{vd} = d$, the Edit distance has to be greater or equal to d. Therefore, all the Edit distances less than d should be covered by $M_{vd} < d$. Thus, M_{vd} also provides a partial ordering of the Edit distances in the transformed space.

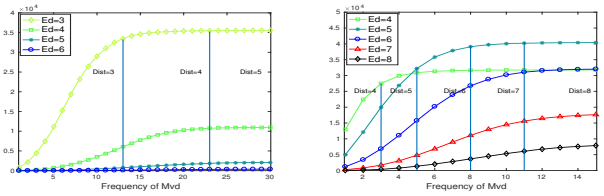
B. Heuristics for Edit distance prediction

We showed that increasing the number of dimensions help predicting the Edit distances in the transformed space based on M_{vd}, specially for smaller Edit distance. Figure 2 shows the prediction accuracy over varying number of reference sequences. It shows that, for $n \geq 300$, the M_{vd}, 1 and 2 will accurately predict the Edit distances 1 and 2, in more than 99% of the cases, which leads to our first heuristic:

H1: For $n \geq 300$, if $M_{vd} \leq 2$, the Edit distance in the transformed space is predicted same as M_{vd}.

We have done experiments with different datasets and found in all these datasets, more than 99% of the cases the prediction is correct. Only in less than 1% of the cases they are not correct but the predicted distances differ by 1.

As we increase M_{vd}, more Edit distances greater than or equal to M_{vd} map into it. Since the reference sequences are far from the database sequences, each component in a VD increases slowly when the corresponding Edit distances become larger. For larger Edit distances, increasing the number of dimensions may not produce an M_{vd} close to the actual Edit distances of pairs of sequence fragments. The predicted M_{vd} will have a range of Edit distances despite having large number of dimensions (Figure 2(d)). Thus, only M_{vd} is not sufficient to satisfactorily predict relatively larger Edit distances.

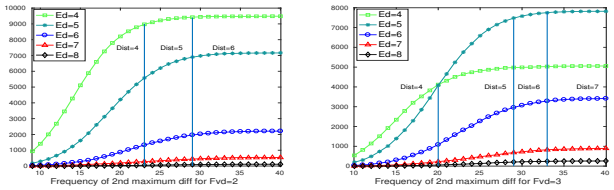


(a) Frequency of $M_{vd} = 3$ (b) Frequency of $M_{vd} = 4$

Fig. 3: Edit distance prediction based on an M_{vd} and F_{vd}s

However, the frequency of M_{vd} is likely to be higher when the actual Edit distance is larger. For example, in Figure 3, cumulative frequencies of Edit distances for given M_{vd} have been shown with the growth of F_{vd}. It also shows the F_{vd} based cutoffs for predicted Edit distances. We select the F_{vd} cutoffs based on the maximum true positive percentage for a given $< M_{vd}, F_{vd} >$. In 3(a), for $M_{vd} = 3$ and $F_{vd} \leq 13$, percentage of correct predictions (sensitivity) and true positive are 97% and 84% respectively. Similarly, we can find F_{vd} cutoffs for predicting higher Edit distances too. Based on this observation, we propose our second heuristic.

H2: For $3 \leq M_{vd} \leq 5$, Edit distances in the transformed space are predicted based on M_{vd} and the corresponding F_{vd}.



(a) $M_{vd} = 4$, $F_{vd} = 2$ (b) $M_{vd} = 4$, $F_{vd} = 3$

Fig. 4: Frequencies of second maximum difference of vector components for given $< M_{vd}, F_{vd} >$

Figure 3(b) presents the predicted Edit distance cutoffs for $M_{vd} = 4$. For $M_{vd} = 4$ and $F_{vd} \leq 3$, sensitivity and true positive rates of predicted Edit distance 4 are 87% and 49% respectively. The accuracy is reduced because with higher M_{vd}, wider range of Edit distances have exactly same M_{vd} and F_{vd}. However, it can be improved by using second maximum difference in a VD. For example, Figure 4 presents the frequencies of second maximum difference in VD for $< M_{vd}, F_{vd} > = < 4, 2 >$ and $< M_{vd}, F_{vd} > = < 4, 3 >$. It is evident from the Figure 3(b) and Figure 4 that for $M_{vd} = 4$, the second maximum difference in VD further enhance the accuracy of the predicted Edit distances.

H2.1: For $4 \leq M_{vd} \leq 5$, second maximum difference and the corresponding frequency in a VD further improves the prediction accuracy.

Although, the predicted Edit distances based on $H2$ and $H2.1$ are more accurate, their error rates are within certain lower bounds. In many cases, the erroneously predicted distances differ by 1-2 points from the actual Edit distances.

However, prediction accuracy declines sharply for even larger M_{vd}. To solve this problem we determine candidate pairs based on maximum two differences of components and their corresponding frequencies in VD. We then compute actual Edit distances for these candidate pairs. Since, we are only interested about clustering sequence within a given distance threshold, the number of candidate pairs is very small compared to the number of pairs in the original Edit space. Utilizing frequencies of second maximum differences of VD further reduces the number of candidate pairs.

H3: For $M_{vd} > 5$, the candidate pairs are selected based on maximum two differences of components and their frequencies in VD. Then, we calculate the Edit distance of those and keep only the pairs which are within a given distance threshold.

C. Optimal Number of Reference Sequences

Increasing number of dimensions improves prediction accuracy in the transformed space, however it increases computation cost. We determine the optimal number of dimensions by minimizing the number of costly Edit distance calculations. First, the number of Edit distance calculation for space transformation increases linearly with the number of dimensions.

Second, with increasing number of dimensions, the number of candidate pairs based on heuristic $H3$ decreases gradually until leveling off for large number of dimensions such as $n = 1000$. It should be noted that the Edit distances of the candidate pairs are measured to verify the actual distance. Figure 5 shows the relationship between the number of Edit distance calculation in the transformed space with the increasing number of dimensions for datasets presented in Table II. The Figure shows similar trend despite the difference of the dataset sizes. Hence, we choose the minimum point in the U-shape curve which gives the optimal number of reference sequences (dimensions).

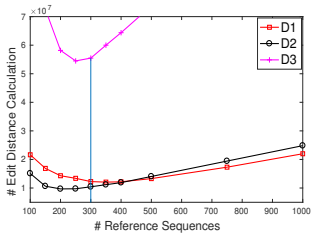


Fig. 5: # Edit distance computation in the transformed space

V. Clustering

We use hierarchical clustering technique, because such techniques are frequently used by biologists. Although many hierarchical clustering techniques exist in literature, we chose a graph based external memory technique denoted by SparseHC, [13] to ensure scalability of our proposed technique.

The proposed technique improves the distance calculation time complexity through space transformation, which is scalable with dataset size. Time complexity of space transformation and pairwise distance calculation in the transformed space are $O(nNl^2)$ and $O(nN^2)$ respectively where n is the number of reference sequence. With the growth of database size, all pairwise distance calculation time in the Edit space increases rapidly compared to that of other components of the total clustering time.

We also predict average linkage cluster thresholds in the transformed space which is shown in Table I. Given, $n \geq 300$ and $l \leq 252$, heuristics $H1$ and $H3$ nearly accurately predict the Edit distances. Although, the prediction accuracy for heuristic $H2$ is comparatively lower than the other heuristics, majority of the erroneous predictions differ by 1 to 2. The predicted thresholds are slightly smaller for 97% and 98% similarity thresholds due to the erroneous predictions of heuristic $H2$. Non-edit distance based methods such k-mer, local sensitive hashing have similarity thresholds. However, these thresholds are not similar to Edit distance based thresholds.

TABLE I: Comparison of clustering thresholds

	Similarity=99%	Similarity=98%	Similarity=97%
Edit Space	2.5	5	7.5
Transformed Space	2.5	4.7	7.3

VI. Results and Discussions

We conduct extensive experiments to evaluate effectiveness of the proposed space transformation based hierarchical clustering. The experiments are performed in an Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz with 24GB physical memory. The linkage criteria is set to average linkage. In the following text, the Edit and the transformed space clustering are also referred to as $Ed - sp$ and $Tr - sp$. Supplementary resources of the paper are available at this link[1].

TABLE II: 16S rRNA sequence fragment datasets

Dataset	Id	# Sequences	Mean Length	St. Dev.
Mouse	D1	17993	252.6	1.233
Human	D2	21575	252.9	1.203
Soil	D3	99742	253.1	1.123
H.pylori Mouse	D4	500000	252.03	2.984

A. Datasets

A diverse set of 16s rRNA sequence fragment datasets generated through high throughput NGS technologies are used. However, there are variations in the sequence lengths because of inserts/deletes and sequencing errors. Table II briefly describes the metagenomic sample datasets, namely mouse, human, soil and H. pylori infected C57/Bl6 mice (SRX1178974 [14]) samples.

TABLE III: Pairwise distance calculation time

Method \ Datasets	D1	D2	D3	D4
Ed-sp	0.614	0.083	0.02	0.021
Tr-sp	0.931	0.068	0.018	0.018

B. Clustering Time

In this section, we evaluate the clustering time performance in the transformed space. First, we compare all pairwise distance calculation time between the Edit and the transformed space. As shown in Table III, the required time in the transformed space is more than an order of magnitude faster than that in the Edit space. For example, Edit distance calculation time of the dataset D4 is more than 15 days while it is only about 11 hours in the transformed space.

Figure 6 shows the comparison of clustering time for different similarity thresholds. Among the compared techniques, only VSEARCH is faster than the proposed space transformation based hierarchical clustering although the clusters accuracy is significantly lower than the proposed technique. Clustering time for both the Edit space based hierarchical technique and the LSH-Div increase more rapidly than the proposed technique for increasing database sizes.

C. Cluster Similarity

We quantify in Table IV the relative similarity of the clusters generated by the proposed technique, VSEARCH and LSH-Div with those of the Edit space hierarchical clustering by using the NMI and the AMI scores. For 99% similarity (Edit distance threshold=2.5), the clusters of the Edit and the transformed spaces are nearly similar because most of the predicted Edit distances in the transformed space are based

[1]https://www.cse.msu.edu/∼islama/edit-distance-prediction/

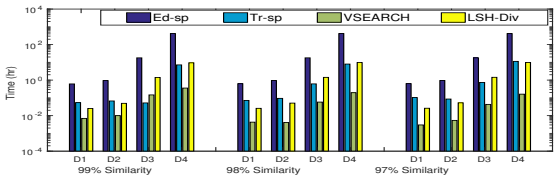


Fig. 6: Comparison of clustering time

TABLE IV: Relative similarity of the clusters

Similarity	Data set	Tr-sp NMI / AMI	VSEARCH [2] NMI / AMI	LSH-Div [1] NMI / AMI
99% (Edit Dist=2.5)	D1	0.998 / 0.993	0.903 / 0.75	0.89 / 0.68
	D2	0.997 / 0.9923	0.89 / 0.77	0.9 / 0.7
	D3	0.9971 / 0.9905	0.91 / 0.78	0.92 / 0.78
	D4	0.9956 / 0.9902	0.887 / 0.774	0.91 / 0.706
98% (Edit Dist=5)	D1	0.982 / 0.9725	0.91 / 0.767	0.88 / 0.702
	D2	0.98 / 0.9745	0.91 / 0.78	0.884 / 0.71
	D3	0.979 / 0.9691	0.92 / 0.79	0.91 / 0.784
	D4	0.981 / 0.969	0.89 / 0.78	0.896 / 0.706
97% (Edit Dist=7.5)	D1	0.9956 / 0.989	0.915 / 0.79	0.87 / 0.732
	D2	0.9938 / 0.9918	0.905 / 0.803	0.87 / 0.76
	D3	0.992 / 0.987	0.93 / 0.82	0.91 / 0.8
	D4	0.992 / 0.9856	0.905 / 0.798	0.89 / 0.72

on heuristic $H1$ ($accuracy > 99\%$). For 98% similarity (Edit distance threshold=5), majority of the predicted Edit distances are based on heuristics $H1$, $H2$ and $H2.1$. Since heuristics $H2$ and $H2.1$ are less accurate, the NMI and AMI scores are relatively low for clusters of 98% similarity. For 97% similarity (Edit distance threshold=7.5), the clusters also include heuristic $H3$ based Edit distances which are correct. Because of the high accuracy of heuristics $H1$ and $H3$, the impact of less accurate heuristics $H2$ and $H2.1$ based predictions is minimized. Hence, the relative similarity scores improve for clusters of 97% similarity than those of 98%. However, the relative similarity of both VSEARCH and the LSH-Div techniques are significantly lower. Although, the NMI scores are around 0.9 for both techniques, the AMI scores are quite low for them.

TABLE V: Silhouette coefficient of the resulting clusters

Similarity	Dataset	Ed-sp	Tr-sp	VSEARCH	LSH-Div
99% (Edit Dist=2.5)	D1	0.314	0.31	0.18	0.15
	D2	0.415	0.41	0.29	0.02
	D3(30k)	0.34	0.334	0.257	0.08
98% (Edit Dist=2.5)	D1	0.426	0.407	0.26	0.14
	D2	0.443	0.42	0.30	0.01
	D3(30k)	0.37	0.352	0.268	0.07
97% (Edit Dist=2.5)	D1	0.57	0.56	0.4	0.12
	D2	0.47	0.46	0.31	-0.084
	D3(30k)	0.41	0.399	0.28	0.05

D. Cluster Quality

We evaluate quality of the clusters using silhouette coefficient. Table V shows that the quality of the clusters generated in the transformed space is very close to that of Edit space, while others are not as close.

VII. Conclusion

In this paper, instead of calculating edit distance in traditional ways, we propose a novel Edit distance prediction technique and use these predicted Edit distances to cluster sequence fragments. We use average linkage hierarchical clustering and show that our method is an order of magnitude faster than the Edit distance based clustering while maintaining atleast 99% accuracy for most of the datasets. The better quality of our clusters is due to the fact that our predicted Edit distances are mostly accurate and where they are not they are within 1-2 points apart. Some theoretical basis for our proposed Edit distance prediction has been provided.

Acknowledgment

The authors would like to thank Dr. James Cole, director of RDP & Dr. Ben Chai of RDP, MSU. The work is partially funded by Collaborative Project with Scientists & Technologists of Indian Origin Abroad Program, Department of Science and Technology, Govt. of India.

References

[1] Z. Rasheed, H. Rangwala, and D. Barbará, "16s rrna metagenome clustering and diversity estimation using locality sensitive hashing," *BMC systems biology*, vol. 7, no. 4, p. S11, 2013.

[2] T. Rognes, T. Flouri, B. Nichols, C. Quince, and F. Mahé, "Vsearch: a versatile open source tool for metagenomics," *PeerJ*, vol. 4, p. e2584, 2016.

[3] S. M. Huse, D. M. Welch, H. G. Morrison, and M. L. Sogin, "Ironing out the wrinkles in the rare biosphere through improved otu clustering," *Environmental microbiology*, vol. 12, no. 7, pp. 1889–1898, 2010.

[4] T. S. Schmidt, J. F. Matias Rodrigues, and C. Mering, "Limits to robustness and reproducibility in the demarcation of operational taxonomic units," *Environmental microbiology*, vol. 17, no. 5, pp. 1689–1706, 2015.

[5] S. L. Westcott and P. D. Schloss, "Opticlust, an improved method for assigning amplicon-based sequence data to operational taxonomic units," *mSphere*, vol. 2, no. 2, pp. e00 073–17, 2017.

[6] J. Venkateswaran, D. Lachwani, T. Kahveci, and C. Jermaine, "Reference-based indexing of sequence databases," in *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 906–917.

[7] P. Papapetrou, V. Athitsos, G. Kollios, and D. Gunopulos, "Reference-based alignment in large sequence databases," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 205–216, 2009.

[8] S. Wandelt, J. Starlinger, M. Bux, and U. Leser, "Rcsi: Scalable similarity search in thousand (s) of genomes," *Proceedings of the VLDB Endowment*, vol. 6, no. 13, pp. 1534–1545, 2013.

[9] A. M. Benjamin, M. Nichols, T. W. Burke, G. S. Ginsburg, and J. E. Lucas, "Comparing reference-based rna-seq mapping methods for non-human primate data," *BMC genomics*, vol. 15, no. 1, p. 570, 2014.

[10] R. C. Edgar, "Search and clustering orders of magnitude faster than blast," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, 2010.

[11] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[12] P. D. Schloss, S. L. Westcott, T. Ryabin, J. R. Hall, M. Hartmann, E. B. Hollister, R. A. Lesniewski, B. B. Oakley, D. H. Parks, C. J. Robinson *et al.*, "Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities," *Applied and environmental microbiology*, vol. 75, no. 23, pp. 7537–7541, 2009.

[13] T.-D. Nguyen, B. Schmidt, and C.-K. Kwoh, "Sparsehc: a memory-efficient online hierarchical clustering algorithm," *Procedia Computer Science*, vol. 29, pp. 8–19, 2014.

[14] "H. pylori infected c57/bl6 mice," 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/sra/SRX1178974[accn]