MeShClust: an intelligent tool for clustering DNA sequences

Benjamin T. James [1,2], Brian B. Luczak [1,2] and Hani Z. Girgis [1*]

[1]Bioinformatics Toolsmith Laboratory, Tandy School of Computer Science, University of Tulsa, 800 South Tucker Drive, Tulsa, OK 74104, USA and [2]Mathematics Department, University of Tulsa, 800 South Tucker Drive, Tulsa, OK 74104, USA

ABSTRACT

Sequence clustering is a fundamental step in analyzing DNA sequences. Widely-used software tools for sequence clustering, such as CD-HIT and UCLUST, utilize greedy approaches that are not guaranteed to produce the best results. These tools are sensitive to one parameter that determines the sequence similarity between sequences in a cluster. Often times, a biologist does not know the exact sequence similarity. Therefore, clusters produced by these tools do not likely match the real clusters comprising the data if the provided parameter is inaccurate. To overcome this limitation, we adapted the mean shift algorithm, which has been used successfully thousands of times in fields such as image processing and computer vision. Here we describe the first application of the mean shift algorithm to clustering DNA sequences. Further, we applied machine learning to devising a novel alignment-free method for measuring sequence similarity. These two innovations represent the core of our software tool, MeShClust. We demonstrate MeShClust's outstanding ability to cluster DNA sequences with high accuracy even when the sequence similarity parameter provided by the user is not very accurate.

INTRODUCTION

Clustering nucleotide sequences is an essential step in analyzing biological data. Pioneering sequence clustering tools have been proposed for reducing redundancy and correcting errors in the next-generation sequencing data (1, 2, 3) and for assembling genomes de-novo (4, 5). Sequence clustering tools were also proposed for barcodes error correction (6) and for taxonomic profiling (7). In addition, CD-HIT (3, 8) and UCLUST (9) are general-purpose sequence clustering tools. These tools are applied to clustering expressed sequence tags, RNA, and reducing a set of sequences to a non-redundant group of sequences.

Until today, some of the most widely-used tools for sequence clustering, such as CD-HIT and UCLUST, depend on greedy algorithms, which are not guaranteed to find the optimal solution. Given the importance of sequence clustering in the computational biology field, we propose a much more advanced approach. The mean shift algorithm is a general-purpose optimization technique (10), which has been widely applied in image processing and computer vision (11, 12, 13). Here, we describe a novel software, MeShClust, utilizing the mean shift algorithm for clustering nucleotide sequences. Further, our adaptation of the algorithm utilizes a novel alignment-free sequence similarity measure.

In practice, the underlying sequence similarity that separates clusters is not known; therefore a biologist has to guess an identity score to provide to the clustering tool. If wrong, this guessed score limits the quality of the predicted clusters markedly. For example, if the provided identity score was higher than the true identity score, a tool would produce smaller clusters; if it was much lower, a tool would produce larger clusters. In both situations, the predicted clusters do not match the real clusters. Further, the related tools are based on greedy algorithms, in which selecting the sequence representing the center of a cluster is not necessarily optimal. In these algorithms, a sequence that does not belong to any cluster is considered the center of a new cluster. Once a center is selected it does not change. To illustrate, if the center sequence is at the periphery of the real cluster, then the predicted cluster is very likely to be a partial cluster. Because the core of MeShClust is the mean shift algorithm, it overcomes these two limitations. Specifically, MeShClust is flexible and is capable of correcting the provided identity score to a great extent. In addition, the sequence representing a cluster does change, moving toward the true center of the cluster. Thus, MeShClust provides a stable clustering algorithm that is not sensitive to the sequence similarity parameter and provides greater accuracy than its counterparts.

MATERIALS AND METHODS

Overview

Algorithms 1 and 2 give an overview of the algorithms underlying our software tool, MeShClust. The software consists of two components. A classifier comprises the first component, which predicts whether or not two sequences are similar to each other. The similarity is measured as the identity score based on the global alignment of the two sequences (14). Sequences are represented as histograms of counts of short words in the sequences. The classifier predicts the similarity between two sequences by calculating a weighted sum consisting of few alignment-free similarity measures using a General Linear Model (GLM). The mean shift algorithm (10) is the core of the second component. Similar to the classifier, the mean shift algorithm processes

*To whom correspondence should be addressed. Tel: +1 918 6313645; Fax: +1 918 6312927; Email: hani-girgis@utulsa.edu

© 2017 The Author(s)

Algorithm 1 An overview of the algorithm implemented in MeShClust

Input: A set of n nucleotide sequences $S=\{s_1,s_2,\ldots,s_n\}$ sorted by decreasing length
Output: Clusters of sequences and their respective centers

Train the classifier using a subset of S to recognize similar sequences to a query sequence
$center_{cur} \leftarrow s_1$
$cluster_{cur} \leftarrow \{s_1\}$

while S is nonempty **do**
  $G \leftarrow$ all sequences from S close to $center_{cur}$ according to the classifier
  $S \leftarrow S-G$
  if G is nonempty **then**
    $cluster_{cur} \leftarrow cluster_{cur} \cup G$
    Run $MeanShift(center_{cur}, cluster_{cur})$ to update $center_{cur}$ (Algorithm 2)
  else
    Add $center_{cur}$ to $Centers$
    Add $cluster_{cur}$ to $Clusters$
    $center_{cur} \leftarrow$ the closest sequence in S to the old $center_{cur}$ according to the Czekanowski similarity (Equation **4**)
    $cluster_{cur} \leftarrow \{center_{cur}\}$
  end if
end while

for i = 1 to 15 **do**
  for all $center_j \in Centers, cluster_j \in Clusters$ **do**
    Run the mean shift to update $center_j$ using sequences in $cluster_j$ along with neighboring clusters (Algorithm 2)
  end for

  for $j=1$ to $|Centers|$ **do**
    for all $center_k \in Centers$ close to $center_j$ **do**
      Merge centers $center_j$ and $center_k$ if $center_j$ and $center_k$ are close
    end for
  end for
end for

Algorithm 2 Sequence clustering using the mean shift

Input: The current center, $center_{cur}$, of a cluster, and a set of points, X
Output: The closest point in X to the new center

Calculate the new center, $center_{new}$, using the current center, $center_{cur}$, according to Equation **1**.

$$center_{new} = \frac{\Sigma_{i=1}^{n} D(x_i, center_{cur}) \cdot x_i}{\Sigma_{i=1}^{n} D(x_i, center_{cur})} \quad \textbf{(1)}$$

Here, D is the classifier (Equation **2**).

$$D(x,q) = \begin{cases} 1 & x \text{ is close to the query sequence, } q \\ 0 & otherwise \end{cases} \quad \textbf{(2)}$$

Next, the closest point $p \in X$ to $center_{new}$ is found using the Czekanowski similarity (Equation **4**), and p is returned as the new center.

the histograms of the input sequences. The mean shift is an iterative, gradient-ascent algorithm that is capable of finding local optimal points. In this adaptation of the algorithm, a local maximum represents the center of a cluster of sequences. In each iteration, a center is recalculated as the weighted *mean* of histograms. This weighted mean is calculated only from the sequences that are similar, as predicted by the classifier, to the center of a cluster. Once updated, a center will *shift* toward a local maximum. As these centers move, some of them converge to the same local maximum; therefore, the algorithm merges them. For this reason, the user do not need to specify the number of centers as opposed to other clustering algorithms such as the k-means. Once the algorithm converges, sequences that contributed to the calculation of a center are considered members of its cluster. Supplementary Data Sets 1–3 contains the source code and the executables of MeShClust.

Next, we give the details of each step of the algorithm. First, we describe how a sequence is represented as a k-mer histogram. Second, the details of the classifier are given. In the third step, the initial clusters are formed. We illustrate the construction of the final clusters in the forth step.

Representing a sequence as a histogram of k-mers

A sequence consists of the nucleotides: A, C, G, and T (or U). A k-mer is a short subsequence of length k. For example, AAA, AAC, AAT, and AAG are called tri-mers. To construct a histogram from a sequence, A, C, G, and T are converted to 0, 1, 2, and 3, and a k-mer is converted to a quaternary number. Horner's rule can be used for calculating the quaternary numbers of a long sequence efficiently (15). The count of a k-mer in the histogram is initialized to 1 instead of 0, these pseudocounts are needed to allow events that "seem" impossible to be able to happen (16). For example, k-mers that are absent from one sequence could be present in another. Pseudocounts are important while calculating conditional probabilities. The selection of this k parameter depends on the size of the input sequences. MeShClust automatically computes the k by first taking the log_4 of the average sequence length, then by subtracting 1. We empirically found that this formula preserves enough information to accurately determine similarity. A smaller k value decreases the amount of memory needed for each histogram and the time required to calculate the alignment-free statistics by a factor of 4 for each nucleotide (17).

Once sequences are converted to k-mer histograms, the classifier is trained in the next stage.

Identifying similar sequences

MeShClust utilizes a classifier to predict similar sequences to a query sequence. The similarity is determined according to an

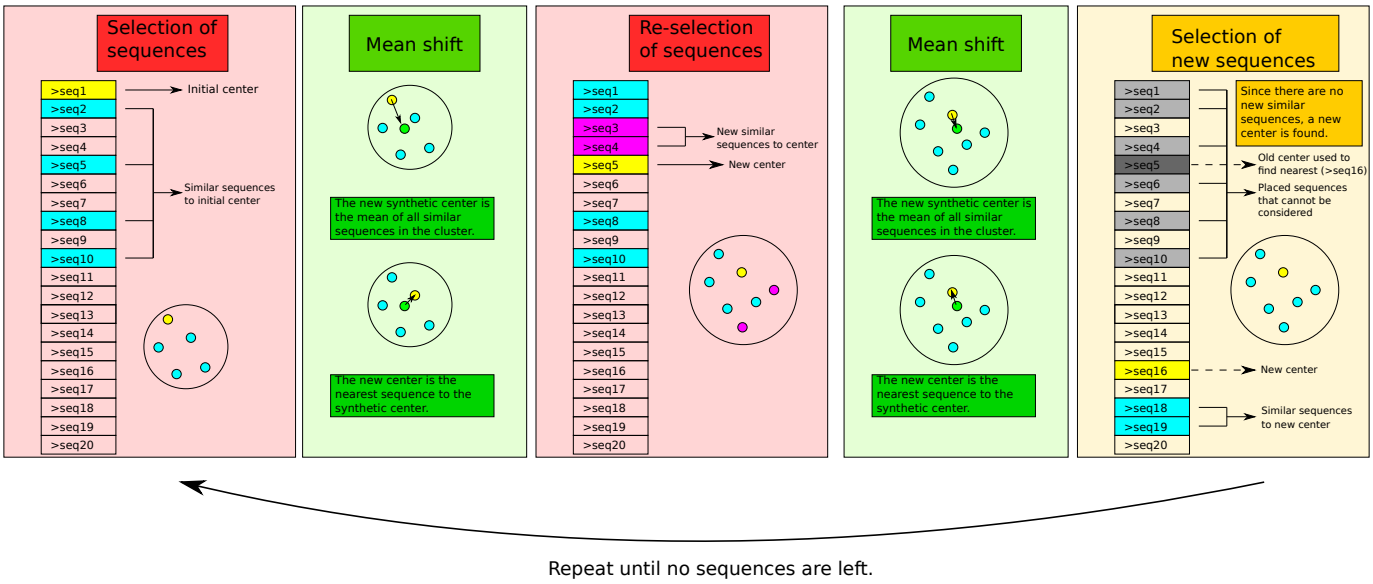

Figure 1. Overview of the initialization step of MeShClust. The diagram shows the iterative process of finding initial clusters of sequences, getting better clusters from the selected sequences, and then finding the next closest cluster.

identity score obtained by a global alignment algorithm (14). With regard to a query sequence, similar sequences can be viewed as one class and dissimilar sequences as the other. Therefore, this task can be represented as a classification task. To this end, we used a GLM (18) for classifying these two classes. As a first step, MeShClust samples a roughly equal number of pairs of similar and dissimilar sequences based on a user-defined cutoff. A large number of pairs of sequences is needed to be sampled. Therefore, about 500 sequence pairs are sampled from each class. Similar sequences are labeled with 1's and the dissimilar sequences with -1's. After that, four features are extracted for each pair of sequences. These four features are selected according to a comprehensive evaluation of alignment-free k-mer statistics (17). The first feature is sequence length difference (Equation **3**) $\times$ Czekanowski similarity (Equation **4**). Length difference $\times$ Jenson Shannon divergence (Equation **5**) represents the second feature. The third and the forth features are Similarity Ratio (Equation **6**), and Squared Chord (Equation **7**).

$$|length(sequence1) - length(sequence2)| \tag{3}$$

$$\sum_{i=0}^{N} \frac{min(A_i, B_i)}{A_i + B_i} \tag{4}$$

$$\frac{1}{2}\sum_{i=0}^{N} P(A_i) log(\frac{2P(A_i)}{P(A_i)+P(B_i)}) + P(B_i) log(\frac{2P(B_i)}{P(A_i)+P(B_i)}) \tag{5}$$

$$\frac{A \cdot B}{A \cdot B + \sqrt{\sum_{i=0}^{N}(A_i - B_i)^2}} \tag{6}$$

$$\sum_{i=0}^{N} A_i + B_i - 2\sqrt{A_i B_i} \tag{7}$$

Here, A and B are the two histograms representing two sequences; P(A_i) and P(B_i) are the probabilities of the ith k-mer in A and B; The "$\cdot$" operator represents the dot product of two vectors. Next, the four features are scaled between 0 and 1 and converted to similarity measures if necessary by subtracting the scaled value from 1. Then MeShClust utilizes an incremental, automatic process to train the GLM. First, it trains the GLM using the first two features. If the training accuracy is at least 97.5%, the training process is finished. Otherwise, it continues by adding another feature followed by evaluating the accuracy. This accuracy is measured by the average of the true positive rate (sensitivity) and the true negative rate (specificity). Once trained, the GLM is used for predicting sequences similar to a query sequence.

At this point, we illustrated how sequences are represented as k-mer histograms and how the classifier is trained. Next, we give the details of forming the initial clusters using the classifier and the mean shift algorithm.

Finding initial clusters

MeShClust aims at clustering the input sequences into distinct groups. Figure 1 diagrams this step. To start, MeShClust gathers similar sequences into initial clusters. This is done by first considering a sequence as the current cluster center, then using the classifier to find similar sequences to the that center. After that, the mean shift is applied on all sequences in the current cluster to calculate the updated mean. Next, the sequence closest to the new mean becomes the center of the cluster. Therefore, at each iteration of the algorithm, a better representative center of the cluster is found, allowing for the addition of more similar sequences to the cluster. This step is repeated until no similar sequences are left. At this point, the current cluster is set aside and a new cluster is formed using the next closest sequence to the last center. The selection of the next center improves clustering by producing a semi-sorted list of sequences; neighboring clusters that may be merged later are grouped near each other. In effect, the combination of using a binary classifier and running the mean shift represents a "flat kernel"(10), expect it only considers sequences not already placed in the initial clusters.

After grouping the sequences into initial clusters, the mean shift is used once again for forming the final clusters.

Clustering

In the previous step, the classifier only considers the unplaced sequences; therefore, some of the initial clusters may have similar sequences in the already placed clusters. Further, the center of a cluster is updated in the initialization step by considering its sequences only. In this step, unlike the initial clustering, the mean shift considers sequences in neighboring clusters. These neighboring clusters include 5 clusters above and below the current cluster. Recall that these clusters are placed in a semi-sorted list. Centers that are close to each other, as determined by the classifier, are merged. If two centers are merged, the sequences that belong to each center are also merged. The algorithm converges if no centers are merged during 3 subsequent iterations.

RESULTS

We start with defining multiple evaluation measures in order to evaluate MeShClust. These measures are intended to evaluate the quality of the predicted clusters as well as the time and the memory requirements of each software tool. After that, we discuss the results of comparing four sequence clustering software tools, including MeShClust.

Evaluation criteria

We used four evaluation measures in evaluating MeShClust and three related tools. Only clusters of at least five sequences are considered in our analysis. The first criterion is the intra-cluster similarity, which is the average similarity between the sequence representing the center of a cluster and the other sequences in the same cluster. Sequence similarity is determined by calculating the identity score. To measure the dissimilarity between different clusters, we applied the inter-cluster similarity measure, which is the average similarity between different centers. Our third criterion is a variant of the Silhouette (19) score. This measure compares the suitability of placing a sequence in its current cluster to the suitability of placing this sequence in the closest cluster. We define $d_c(s)$ as the distance between the sequence, s, and the sequence representing its own cluster and $d_n(s)$ as the distance between s and the sequence representing the closest "neighbouring" cluster. The distance is measured by subtracting the identity score from 100. Equation **8** defines the Silhouette score.

$$Silhouette = \frac{1}{n} \sum_{s \in sequences} \frac{d_n(s) - d_c(s)}{\max(d_n(s), d_c(s))} \quad \textbf{(8)}$$

Here, n is the number of sequences in the set. The Silhouette score ranges between 1 and -1; the higher, the better. Our final evaluation measure is the one to one criterion, which is applied when the true clusters are known. A predicted cluster matches a real cluster if at least 80% of its sequences belong to a real cluster. The one to one is the percentage of the matched clusters.

Next, we apply these evaluation measures to assessing the performance of MeShClust and three widely used tools.

Comparison to related tools

We utilized both synthetic and real data sets in comparing the performance of MeShClust to the performances of UCLUST, CD-HIT, and DNACLUST (7), which are widely used clustering tools. To start, we generated three data sets, which we call the 10%, the 25%, and the 100-centers sets (Supplementary Data Set 4–6). The three synthetic data sets were generated at 10%, 25%, and 25% mutation rates, respectively. Each of the 10% and the 25% data sets has 10 clusters, each of which consists of about 25 sequences. The 100-centers set has 100 clusters each of size 10. The length of a sequence is 1000 base pair approximately.

To test the tolerance of the four tools to the identity parameter, we ran the tools using five identity scores (up to 10% above and below the actual identity score). Two of these scores are above the real identity score, and two are below it, and one approximately matches the identity score used for generating the clusters. These identities were selected to demonstrate to what degree a tool is tolerant to an inaccurate identity score. Table 1 shows the results of evaluating the four tools on the 10% data set using these identity scores: 75%, 80%, 85%, 90% and 95%. The true identity score is between 85% and 90%. On the synthetic data sets, the performance is best measured by the one to one percentage because the true clusters are known. On four out of the five tests, MeShClust obtained perfect one-to-one scores. Among the three related tools, UCLUST obtained perfect one-to-one score in one test only. CD-HIT and DNACLUST did not obtain perfect or close to perfect results on any of the five tests. Further, MeShClust achieves much better results in terms of the Silhouette, the intra-cluster similarity scores. Similar results were obtained on the 25% and the 100-centers data sets (see Supplementary Table 1). These results demonstrate that MeShClust has great tolerance to inaccurate identity score. This tolerance is evident by the consistency of the high quality clusters obtained on different data sets at different identity levels.

Next, we aimed at evaluating the tools on real data; therefore, we obtained sequences from a microbiome study (20). We call this set the Costello set. About 1.1 million sequences comprise this set. Sequences in this set range between 200 and 400 base pairs. Clusters are defined using 97% identity score. Because the real clusters are unknown, we generated similar synthetic sets. We generated the 15K and the 150K sets consisting of 15 and 150 thousands sequences, respectively (Supplementary Data Sets 7 and 8). A synthetic cluster contains between 50 and 100 sequences. These clusters were generated using 3% mutation rate; however, the actual mutation is usually higher than 3% due to randomization.

As before, we ran five tests using the following identity scores: 83%, 87%, 90%, 93%, and 97%. On the 15K set, MeShClust obtained perfect and CD-HIT obtained almost perfect one-to-one percentage in four tests, whereas UCLUST obtained 92 and 81 one-to-one on two sets; DNACLUST preformed well in one test only. MeShClust achieved similar, even better, results on the 150K data set. Specifically, it obtained 100% one-to-one in four tests and 80% in the fifth test. In all the tests when the performances of CD-HIT and MeShClust were comparable, MeShClust obtained higher Silhouette scores, indicating the better quality of MeShClust's

Table 1. Comparison of the performances of MeShClust, UCLUST, CD-HIT, and DNACLUST on the 10%-mutation-rate synthetic data set.

Tool	Identity	One to One	Silhouette	Intra	Inter	Real time (min:sec)	Max memory (MB)
MeShClust	0.75	100%	0.830	90.967	50.509	0:05.66	54.0
UCLUST	0.75	100%	0.704	84.051	50.136	0:00.16	4.4
DNACLUST	0.75	0%	–	–	–	0:04.38	396.1
CD-HIT	0.75	–	–	–	–	0:00.01	2.5
MeShClust	0.80	100%	0.830	90.967	50.501	0:05.87	53.3
CD-HIT	0.80	50%	0.329	84.524	50.084	0:00.26	34.9
UCLUST	0.80	10%	0.400	85.998	50.298	0:00.29	5.9
DNACLUST	0.80	0%	–	–	–	0:02.93	392.0
MeShClust	0.85	100%	0.830	90.967	50.517	0:05.81	53.6
CD-HIT	0.85	10%	0.289	85.030	50.089	0:00.25	35.1
DNACLUST	0.85	0%	–	–	–	0:01.89	392.0
UCLUST	0.85	0%	–	–	–	0:00.53	8.5
MeShClust	0.90	100%	0.830	90.967	50.506	0:05.78	54.0
DNACLUST	0.90	0%	–	–	–	0:01.03	392.0
UCLUST	0.90	0%	–	–	–	0:00.32	8.6
CD-HIT	0.90	0%	–	–	–	0:00.99	38.7
MeShClust	0.95	0%	–	–	–	0:05.68	53.7
UCLUST	0.95	0%	–	–	–	0:00.19	8.7
CD-HIT	0.95	0%	–	–	–	0:00.14	39.0
DNACLUST	0.95	0%	–	–	–	0:00.41	392.0

The 10% data set contains 10 clusters, each of which is generated by mutating approximately 10% of the bases comprising a template sequence. Each cluster consists of 25 sequences. Clusters containing less than 5 sequences are not considered in the evaluation; results labeled with "–" are due to these small clusters. The Silhouette score ranges between -1 and 1. The higher the one to one, the Silhouette, and the intra-cluster scores, the better; the lower the inter-cluster similarity, the better. MeShClust was the only tool that was capable of finding the correct clusters in four tests, whereas UCLUST succeeded in finding the correct clusters in one test only.

Table 2. MeShClust and three related tools were used for clustering the Costello data

Tool	Identity	Silhouette	Intra	Inter	Real time (min:sec)	Max memory (MB)
MeShClust	0.83	0.348	89.150	65.459	19:28.00	3156.1
UCLUST	0.83	0.267	85.741	64.933	02:25.00	412.4
CD-HIT	0.83	0.172	81.821	64.196	10:25.00	491.5
DNACLUST	0.83	0.136	81.875	64.460	02:26.00	1127.3
MeShClust	0.87	0.369	90.751	65.885	25:56.00	3161.0
UCLUST	0.87	0.271	88.269	65.624	03:36.00	413.3
CD-HIT	0.87	0.209	85.546	65.202	10:28.00	492.2
DNACLUST	0.87	0.177	85.671	65.342	03:30.00	1123.4
MeShClust	0.90	0.417	92.199	66.383	23:58.00	3159.2
UCLUST	0.90	0.256	90.227	66.314	01:09.00	412.3
CD-HIT	0.90	0.203	87.704	65.755	12:04.00	493.8
DNACLUST	0.90	0.140	87.541	65.908	04:24.00	1120.1
MeShClust	0.93	0.442	93.642	66.659	29:23.00	3157.1
UCLUST	0.93	0.204	92.107	67.153	08:06.00	412.4
DNACLUST	0.93	0.132	89.952	66.643	01:34.00	1125.9
CD-HIT	0.93	0.180	89.839	66.475	09:19.00	497.8
MeShClust	0.97	0.454	95.614	67.188	28:48.00	3155.3
CD-HIT	0.97	0.155	93.767	67.496	17:31.00	520.6
DNACLUST	0.97	0.114	94.048	67.743	02:48.00	1123.4
UCLUST	0.97	0.055	95.543	67.993	07:44.00	416.0

The Costello data set was obtained from a microbiom study (20). About 1.1 million sequences comprise this data set. Sequence lengths range between 200 and 400 base pairs. The performances were compared using these five identity scores: 83%, 87%, 90%, 93%, and 97%. MeShClust achieved the highest Silhoutte scores and the highest intra-cluster sequence similarity scores in the five tests, demonstrating its excellent performance.

clusters. Supplementary Table 1 shows the results on the 15K and the 150K sets.

After that, we evaluated the four tools on the Costello set (See Table 2). MeShClust obtained higher Silhouette and intra-cluster similarity scores than the three related tools in

the five tests, demonstrating the better quality of the clusters it produces. MeShClust takes longer time than the other tools. However, these additional minutes are worth waiting for given the excellent quality of the clusters produced by MeShClust.

CONCLUSION

DNA sequence clustering algorithms have many applications. Nonetheless, the widely applicable tools depend on a greedy algorithm, which does not necessarily produce the best results. Our clustering software, MeShClust, is a novel tool that utilizes an instance of unsupervised machine learning algorithms, known as the mean shift algorithm. *MeShClust is the first application of the mean shift to clustering DNA sequences*. Further, most sequence clustering tools use a slow quadratic algorithm for sequence alignment. In contrast, MeShClust uses a novel, machine-learning-based, alignment-free method for computing sequence similarity. Furthermore, *this is the first attempt to formulate the task of identifying similar sequences as a classification task*. When tested on multiple synthetic and real datasets, MeShClust outperformed the related tools with a clear margin, significantly advancing the field of sequence analysis.

SUPPLEMENTARY DATA

The C++ source code and the executables of MeShClust and the synthetic data sets are available online as Supplementary Data Sets 1–8 and Supplementary Table 1.

ACKNOWLEDGEMENTS

We are thankful to the anonymous reviewers for taking the time to review this manuscript.

FUNDING

This research was supported by internal funds provided by the College of Engineering and Natural Sciences and the Faculty Research Grant Program at the University of Tulsa. The research results discussed in this publication were made possible in part by funding through the award for project number PS17-015, from the Oklahoma Center for the Advancement of Science and Technology.

Conflict of interest statement. None declared.

REFERENCES

1. Bao, E., Jiang, T., Kaloshian, I., and Girke, T. (2011) SEED: efficient clustering of next-generation sequences. *Bioinformatics,* **27**(18), 2502.
2. Chong, Z., Ruan, J., and Wu, C.-I. (2012) Rainbow: an integrated tool for efficient clustering and assembling RAD-seq reads. *Bioinformatics,* **28**(21), 2732.
3. Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics,* **28**(23), 3150.
4. Warren, R. L., Sutton, G. G., Jones, S. J. M., and Holt, R. A. (2007) Assembling millions of short DNA sequences using SSAKE. *Bioinformatics,* **23**(4), 500.
5. Shimizu, K. and Tsuda, K. (2011) SlideSort: all pairs similarity search for short reads. *Bioinformatics,* **27**(4), 464.
6. Zorita, E., Cusc, P., and Filion, G. J. (2015) Starcode: sequence clustering based on all-pairs search. *Bioinformatics,* **31**(12), 1913.
7. Ghodsi, M., Liu, B., and Pop, M. (2011) DNACLUST: accurate and efficient clustering of phylogenetic marker genes. *BMC Bioinformatics,* **12**(1), 271.
8. Li, W. and Godzik, A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics,* **22**(13), 1658.
9. Edgar, R. C. (2010) Search and clustering orders of magnitude faster than BLAST. *Bioinformatics,* **26**(19), 2460–2461.
10. Cheng, Y. (1995) Mean shift, mode seeking, and clustering. *IEEE Trans Pattern Anal Mach Intell,* **17**(8), 790–799.
11. Comaniciu, D. and Meer, P. (1999) Mean Shift Analysis and Applications. In *Proc IEEE Int Conf Comput Vis* pp. 1197–1203.
12. Comaniciu, D. and Meer, P. (May, 2002) Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans Pattern Anal Mach Intell,* **24**(5), 603–619.
13. Girgis, H. Z., Mitchell, B. R., Dassopoulos, T., Mullin, G., and Hager, G. (April, 2010) An intelligent system to detect Crohn's disease inflammation in Wireless Capsule Endoscopy videos. In *Proc IEEE Int Symp Biomed Imaging* pp. 1373–1376.
14. Needleman, S. B. and Wunsch, C. D. (1970) A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *J Mol Biol,* **48**, 443–453.
15. Girgis, H. Z. (Jul, 2015) Red: an intelligent, rapid, accurate tool for detecting repeats de-novo on the genomic scale. *BMC Bioinformatics,* **16**(1), 227.
16. Compeau, P. and Pevzner, P. (2015) Bioinformatics Algorithms: An Active Learning Approach, Active Learning Publishers, .
17. Luczak, B. B., James, B. T., and Girgis, H. Z. (2017) A Survey and Evaluations of Histogram-Based Statistics in Alignment-Free Sequence Comparison. *Brief Bioinform,* Revisions have been requested.
18. McCullagh, P. (1984) Generalized linear models. *Eur J Oper Res,* **16**(3), 285–292.
19. Rousseeuw, P. J. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math,* **20**, 53 – 65.
20. Costello, E. K., Lauber, C. L., Hamady, M., Fierer, N., Gordon, J. I., and Knight, R. (2009) Bacterial Community Variation in Human Body Habitats Across Space and Time. *Science,* **326**(5960), 1694–1697.