Study of Biological Sequence Clustering
[Internal Report]

Saliya Ekanayake
School of Informatics and Computing,
Indiana University
sekanaya@cs.indiana.edu

ABSTRACT

Determination of biologically related clusters of sequences is important bioinformatics analyses. The similarity between sequences is generally assessed based on their alignments with one another. This could be used with a clustering algorithm to determine groups of sequences, yet it is not straightforward how to get reliable results. We present the factors affecting the quality of clusters and how visualization aids in the refinement of results. We also present a way to verify clusters in the presence of consensus sequences, and represent clusters.

Keywords

Sequence alignment, multi-dimensional scaling

1. INTRODUCTION

The work on biological sequence clustering is to identify the similar biological sequences and to present them in a comprehensible manner to the biologists. This involves a series of steps starting from finding a measure of similarity between sequences to finally presenting a three dimensional view of the similar groups. It is important in this pipeline to capture and preserve the inherent similarity between sequences in order to yield reliable clusters at the end. This requires understanding the effect of different choices available at each step in order to minimize distortions and verifying clusters if possible with existing consensus sequences. Also, in an engineering aspect, implementing some of the steps requires adopting parallel solutions to meet with demanding computational power. In this paper we present our experience and findings over two separate, yet similar, sequence clustering projects involving around million sequences each.

2. SIMPLE ARCHITECTURE

The series of steps involved in sequence clustering is put in a simple pipeline as shown in Figure 1 where numbered items are as follows.

Processes:	Data:
P1 – Pairwise distance calculation	D1 – Input sequences
	D2 – Distance matrix
P2 – Multi-dimensional scaling	D3 – Three dimensional coordinates
P3 – Pairwise clustering	D4 – Cluster mapping
P4 – Visualization	D5 – Plot file

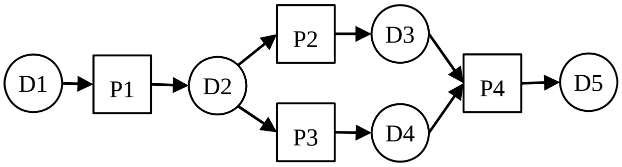


Figure 1. Simple pipeline of steps

Given a set of sequences, the first step is to perform pairwise alignment and determine the similarity between each pair of sequences. Similarity is presented as a distance such that high similarity means small distance and vice versa. The multidimensional scaling program operates on the computed distance matrix and produces a set of three dimensional coordinates to represent sequences while preserving the distance between them. The pairwise clustering program processes the same distance matrix and produces a mapping of sequences to similar groups. The set of coordinates and the cluster mapping is processed together by the visualization program to produce the three dimensional plot of sequences colored into groups found by clustering program.

3. DETERMINATION OF CLUSTERS

The clusters we find are meant to capture the natural closeness of biological sequences, strictly speaking, but there is no definitive way to determine if one cluster mapping is biologically more accurate over the other. Thus, our options are to understand the different factors that influence cluster results and to tune them such that any biological similarity present in the input sequences may get preserved through the steps in the pipeline.

3.1 Visualization to Complement Clustering

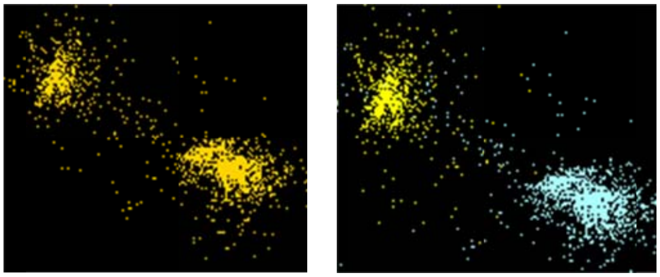
Refinement of clusters would solely depend on a mathematically computed goodness measure if clustering were to be performed without visualization. The caveat is that even though the clustering algorithm is properly written and produces good clusters, it may still fail in discovering proper clusters.

Figure 2 shows a portion of the sequences visualized as points in three dimensions. Colors indicate clusters found by the clustering program. The left figure shows how the clustering program has grouped two seemingly distinct sequence groups into one cluster. Situations like this are common when clustering a large number of sequences as the program would converge satisfying a condition global to all the sequences yet suboptimal for sequences in some clusters. The figure on right shows the effect of clustering sequences only in these two groups. It has resulted two clusters as one would expect.

The reverse of the above scenario, i.e. unnecessary splitting of seemingly well-defined cluster, could also happen. Thus, it is clear how visualization aids to identify these mishaps and correct them by further clustering of selected sections or regrouping of splits as necessary.

3.2 Cluster Size

The number of points in a cluster is an important factor as too many will tend to group more than one structure as a cluster while too little may split actual clusters. There is no deterministic method to know this or the number of clusters in advance. Therefore, we use a hierarchical clustering approach with guidance from biologists on the estimated number of clusters. We also rely on the visual structure produced with multi-dimensional scaling.



(a) Multiple groups identified as one cluster (b) Refined clusters to show proper split of groups

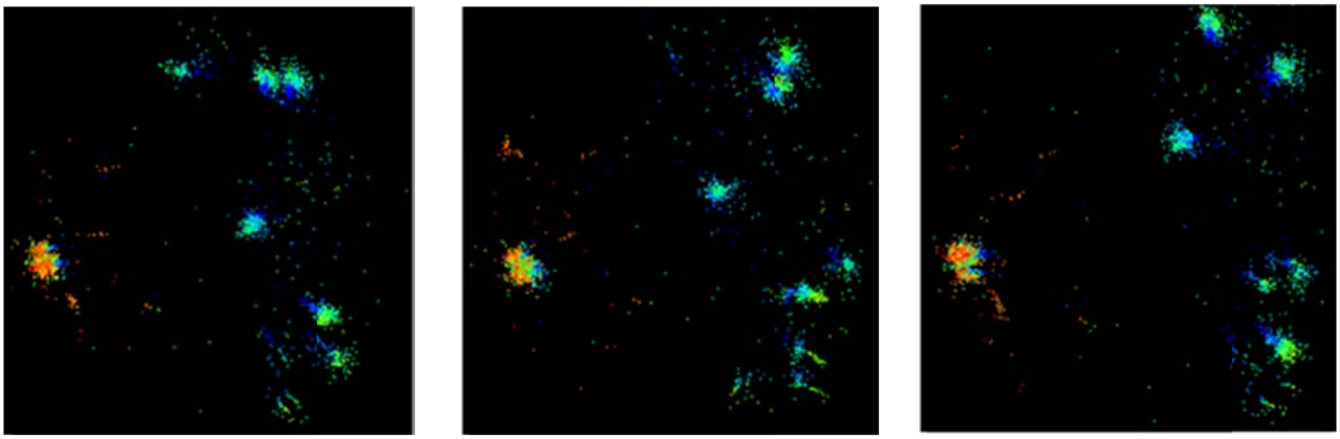
Figure 2. Cluster refinement with the aid from visualization

```
Sequence 1          TTGAGTTTTAACCTTGCGGCCGTA
Sequence 2          AAGTTTCTTGCCGG

                    TTGAGTTTTAACCTTGCGGCCGTA
Global alignment       ||||||   |||   ||||
                    ---AAGTTT---CTT---GCCG–G

                    ttgagttttaacCTTGCGGccgta
Local alignment                 |||||||
                          aagtttCTTGCGG
```

Figure 3. Global vs. local sequence alignment



(a) Gap open -10, gap extension -4 (b) Gap open -16, gap extension -4 (c) Gap open -4, gap extension -4

Figure 4. Comparison of results for different gap penalties

For example, we look at the three-dimensional projection to identify the number of coarse-grained clusters. Usually we could see around 10-15 clusters in our datasets. We use this number to drive the initial clustering. If some of the clusters appear to be a collection of smaller clusters, we will further cluster only those. Figure 5 for example, shows a coarse-grained cluster on left and its refinement on right (note colors distinguish clusters). This technique enables us to get clustering result to agree with the geometrical structure of sequences. It is worth mentioning here that determining if clustering results and geometrical structure agree with actual biological structure is a separate task as discussed in section 4.

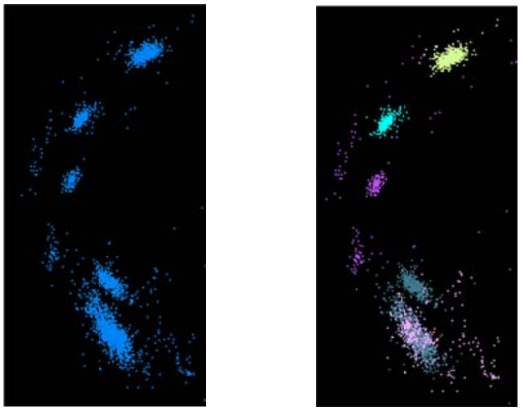


Figure 5. Hierarchical clustering with aid from visualization

3.3 Effect of Gap Penalties

Sequence alignment may insert gaps when no non-gap character pair could be found to yield a better score for the alignment. The decision depends on the penalties associated with introducing and extending gaps, which are generally known as gap penalties. It is possible to get different alignments based on these penalties and in turn result different distances between sequences.

We studied the effect of gap penalties with a smaller 16S rRNA (section 6) dataset of 6822 sequences where the recommended penalties were -16 for gap open (GO) and -4 for gap extension (GE). We tested for the combinations of gap penalties in **Error! Reference source not found.** and found the clustering to show little or no effect compared to the reference case.

Table 1. Combinations of gap penalties

					Ref.									
GO	-4	-4	-8	-10	**-16**	-16	-16	-20	-20	-20	-24	-24	-24	-24
GE	-2	-4	-4	-4	**-4**	-8	-16	-4	-8	-16	-4	-8	-16	-20

A comparison of the results for combinations -4/-4 and -10/-4 against the reference is given below in Figure 4. The number of clusters is the same in all three cases and except for minor differences in shapes and positions the results seem identical. Moreover, the actual position of clusters is also irrelevant as long as they maintain their relative distances, which happened to be true in these tests.

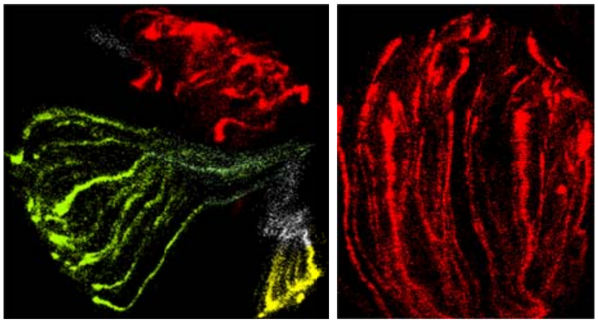


Figure 6. Long thin line formation

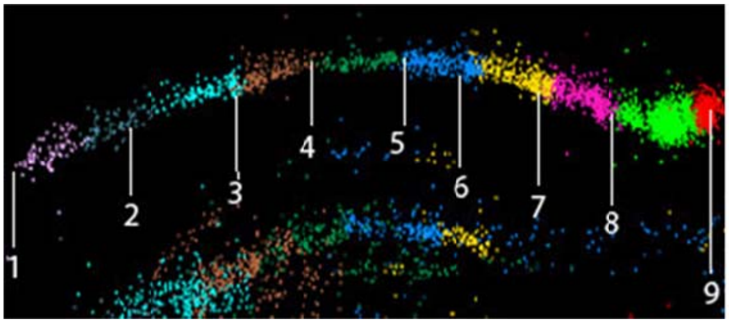


Figure 7. Single line analysis

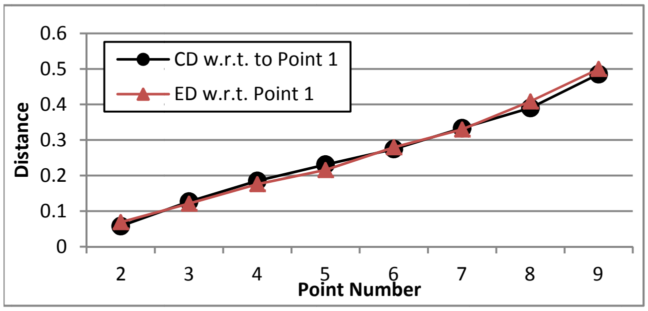


Figure 8. Computed Vs. Euclidean distance

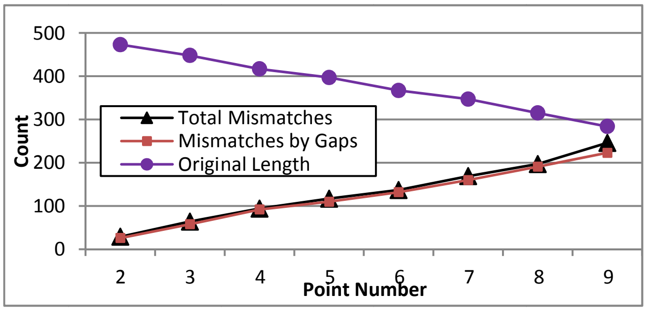


Figure 9. Effect of gaps on the linear formation

3.4 Global Vs. Local Sequence Alignment

The two best known sequence alignment algorithms are Smith-Waterman [1] and Needleman-Wunsch [2], which perform local and global alignment of sequences respectively. Implementation wise both these perform a similar kind of computation. The distinction, however, comes from the fact that Needleman-Wunsch is constrained to find an optimal alignment from end-to-end whereas Smith-Waterman is relaxed to find a subsection producing an optimal alignment.

Figure 3 shows the results of global and local sequence alignment on two sample sequences. The global alignment contains all the characters of both sequences plus gaps giving an alignment of length equal or greater than the length of the longest sequence. The local alignment, in contrast, includes only those characters in upper case (others are shown for clarity of the position).

The decision of which type of alignment to perform, as we experienced, depends on the sequence set and the type of distance measure. For example, the data we used had non uniform length distributions with a range of 181 to 585 in one set and a range of 200 to 1000 in the other. Also, the majority of the smaller sequences had most of their characters as a subset in one or more longer sequences. This nature resulted superfluous alignments with many gaps with Needleman-Wunsch global aligner. The effect was visible both in clustering and visualization results, yet being prominent in the latter as long thin lines.

The two images in Figure 6 show the overall shape resulted for 100,000 sequences of 16s rRNA when mapped into three dimensional points where distances computed based on global alignment. The points have formed long thin line structures, which suggested the Euclidean distances along a line have a linear relationship. We carried two analyses to determine if this was the result of an anomaly or the very nature of biological similarity present in sequences.

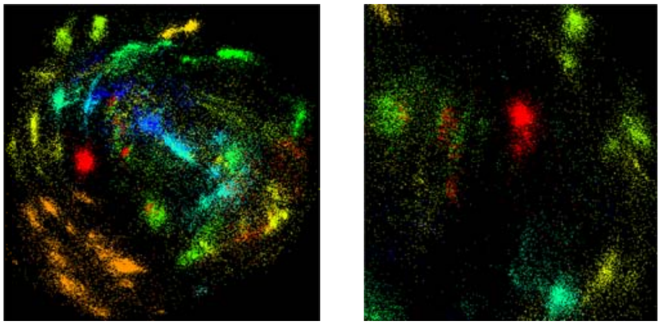
We isolated a line of points and selected nine points in near equal distances along its length as shown in Figure 7. The first analysis compared the computed distance (CD) based on alignments from point 1 to all the other points with the corresponding Euclidean distances (ED) as in Figure 8.

The overlap of lines CD and ED in Figure 8 indicates that the multi-dimensional scaling program has found an accurate set of coordinates to represent sequences, which preserves the original distances between them. The second analysis compared mismatches against the sequence length along the line as in Figure 9. A mismatch is when a character in one sequence is aligned with a different character in the other sequence or a gap. The graph shows a linear increase of the number mismatches while the sequence lengths decrease linearly. The distance measure we computed over alignments has a linear relationship with number of mismatches. Thus Figure 8 and Figure 9 suggest that the linear decrease of sequence length has caused the linear increase in distance due to the increase number of mismatches. Moreover, virtually all mismatches have been caused by gaps as seen in Figure 9. In conclusion, global alignment has introduced gaps dearly to leverage the length difference causing unreliable distance values.

Smith-Waterman local aligner, in contrast, found better alignments with reasonable alignment lengths leading to more spherical shaped clusters than linear ones as shown in Figure 10.

3.5 Distance Types

The distance between two sequences is a computed value based on their alignment that represents the biological closeness between them. Different distance values may be computed over the same alignment depending on the particular choice of distance type. In our experiments we used the well-known Percent Identity (PID) distance type due to the interest of biologists. Additionally, we list few other types that we are currently evaluating in our ongoing research work.



(a) Visualization of 100,000 sequences

(b) Zoomed-in subset of (a)

Figure 10. Multi-dimensional scaling with Smith-Waterman local alignments

Figure 11 shows a general sequence alignment with possible end gaps (note a local alignment will not result end gaps). We name the region excluding end gaps as the aligned region. Pairs of boxes with the same color indicate a match and others indicate mismatches. Pairs with one box and one dash indicate a character being aligned with a gap.

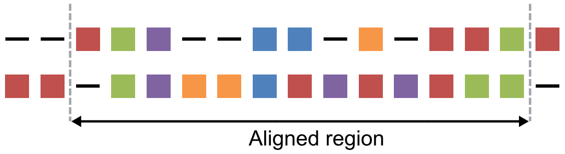


Figure 11. Sequence alignment

3.5.1 *Percent Identity (PID) Distance*

Given the alignment between two sequences, let the number of matching pairs in aligned region be N and the total number pairs in the aligned region be L. The PID distance, δ_{PID}, is then computed according to Eq. 1.

$$\delta_{PID} = 1.0 - \left({}^{N}/_{L}\right) \qquad \text{Eq. 1}$$

3.5.2 *Normalized Score*

Both Needleman-Wunsch and Smith-Waterman algorithms find determine the best alignment between two sequences by maximizing a value called score, S, which is computed as in Eq. 2.

$$S = \sum_{i=1}^{L} s_i \qquad \text{Eq. 2}$$

In Eq. 2 L is the length of the alignment (including end gaps, if any) and s_i is the score for i^{th} pair of aligned characters, which is determined by the scoring matrix and gap penalties. The characters in a sequence belong to a particular alphabet and the scoring matrix contains a score for each possible pairs of such characters. Thus, if i^{th} pair contains two characters then s_i is equal to the score in scoring matrix for that pair. If one of the characters is a gap and the character in the $(i-1)^{th}$ pair of the same sequence is not a gap then s_i is the value of gap open penalty. If the previous character is also a gap then s_i is taken as gap extension penalty.

Figure 12 illustrates the score computation for a hypothetical alignment using the scoring matrix on the left and specified gap penalties.

The score, though represent a similarity measure between the particular two sequences, may not be used directly as a distance measure to compare a set of sequences.

	A	T	C	G
A	5	-4	-4	-4
T	-4	5	-4	-4
C	-4	-4	5	-4
G	-4	-4	-4	5

GO = -16 GE = -4

T	C	A	A	C	C	A	-
T	T	-	-	-	C	T	G
5	-4	-16	-4	-4	5	-4	-16

$$S = 5 + (-4) + (-16) + (-4) + (-4) + 5 + (-4) + (-16) = -38$$

Figure 12. Score of an alignment

However, it is logical to consider the option of using score as a distance since the alignment algorithm finds alignments that optimize this value. As a solution one could use normalized score and we present five normalizations in Eq. 3,4,5,6, and 7.

- Average Local

$$\delta_{AvgLocal} = 1.0 - \left(\frac{S_{ij}}{Avg\left(S_{i'i'} + S_{j'j'}\right)}\right) \qquad \text{Eq. 3}$$

In Eq. 3, S_{ij} is the score for the alignment of sequences i and j.The portion of the sequence i that participates in the alignment is denoted as i' and $S_{i'i'}$ is the score for alignment of i' with itself. $S_{j'j'}$ is similar to $S_{i'i'}$ and Avg is the average function.

- Min Local

$$\delta_{MinLocal} = 1.0 - \left(\frac{S_{ij}}{Min\left(S_{i'i'} + S_{j'j'}\right)}\right) \qquad \text{Eq. 4}$$

Eq. 4 is similar to Eq. 3 except Min represents minimum instead of average.

- Max Local

$$\delta_{MaxLocal} = 1.0 - \left(\frac{S_{ij}}{Max\left(S_{i'i'} + S_{j'j'}\right)}\right) \qquad \text{Eq. 5}$$

Here we take the maximum of $S_{i'i'}$ and $S_{j'j'}$. The rest is as same as in Eq.3 and Eq. 4.

- Min Global

$$\delta_{MinGlobal} = 1.0 - \left(\frac{S_{ij}}{Min\left(S_{ii} + S_{jj}\right)}\right) \qquad \text{Eq. 6}$$

In contrast to equations above, here we consider the self-aligned score of full sequences i and j as indicated by S_{ii} and S_{jj}.

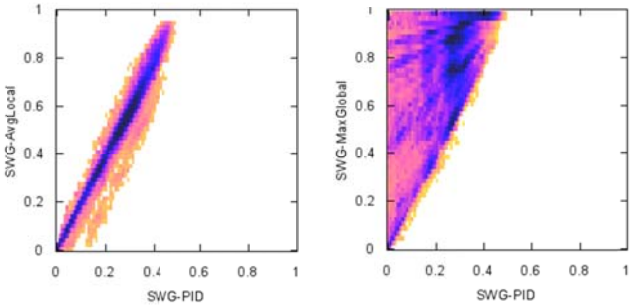
- Max Global

$$\delta_{MaxGlobal} = 1.0 - \left(\frac{S_{ij}}{Max\left(S_{ii} + S_{jj}\right)}\right) \qquad \text{Eq. 7}$$

This is similar to Eq. 5 except we use maximum of self-aligned scores instead of minimum.

Note. These equations are valid for both local and global alignments, yet with global alignment sequences i and j coincide with subsequences i' and j' respectively thereby reducing Eq. 4 to Eq. 6 and Eq. 5 to Eq. 7. Also, $\delta_{AvgLocal}$ may then be termed as $\delta_{AvgGlobal}$.

We studied the difference of these distances with respect to each other and δ_{PID} for local alignment of Fungi [section 6] sequences. The δ_{PID} serves as a reference since we could obtain reasonable clustering results based on it. The comparison indicate that $\delta_{AvgLocal}$, $\delta_{MinLocal}$, and $\delta_{MaxLocal}$ correlate well with each other and δ_{PID}. Figure 13 (a) shows the correlation of $\delta_{AvgLocal}$ and δ_{PID} as a "heatmap". The $\delta_{MinLocal}$ and $\delta_{MaxLocal}$ distances followed similar correlation diagrams with δ_{PID}. The distances based on global self-aligned scores, i.e. Eq. 6 and Eq. 7, however, did not correlate well with each other or with δ_{PID}. Figure 13 (b) shows the correlation of $\delta_{MaxGlobal}$ and δ_{PID}, and $\delta_{MinGlobal}$ followed the same pattern. Details of this analysis are available at [3].



(a) $\delta_{AvgLocal}$ Vs. δ_{PID} (b) $\delta_{MaxGlobal}$ Vs. δ_{PID}

Figure 13. Correlation of normalized score distance versus percent identity distance

3.5.3 Normalized Bit Score

Bit score, S', is a log scaled variant of the raw score, S, computed in Eq. 2. It is used mainly in the popular protein sequence aligner, BLAST [4] and is computed according to Eq. 4.

$$S' = \frac{\lambda S - \ln(K)}{\ln(2)} \qquad \text{Eq. 8}$$

In Eq. 4, S is the raw score computed in Eq 2. Values λ and K are statically determined for the given scoring matrix and gap penalties [5]. The bit score value, unlike raw score, is thus comparable among different alignments. However, we apply the same normalization as in Eq. 3 to compute the normalized bit score distance.

3.6 Statistical Significance

Statistical significance of an alignment indicates how probable it is for such an alignment to happen by chance [6]. For a given two sequences x and y this could be thought of as the probability that a particular score,s, or higher would occur when x and y are randomly shuffled and aligned. We could, thus, improve the distances computed previously by weighting them with the statistical significance.

3.7 Distance Transformation

δ_{11}	δ_{12}	$\cdots$	δ_{15}
δ_{21}	δ_{22}	$\cdots$	δ_{25}
$\cdots$	$\cdots$	$\cdots$	$\cdots$
δ_{51}	δ_{52}	$\cdots$	δ_{55}

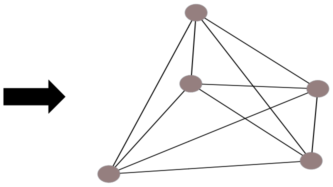


Figure 14. Three dimensional mapping of distances

Given a distance matrix, multi-dimensional scaling would find a set of points that preserves the pairwise distances as shown in Figure 14. However, the distances often come from higher dimensions and reducing them to three dimensional points may produce "less reasonable" result where higher dimensional points are concentrated to the edge of the surface.

As a solution, we are evaluating different mapping functions that would reduce the dimensionality of the input distances to multi-dimensional scaling program. These mapping functions are monotonic and satisfy $\forall \delta_1, \delta_2 : \delta_1 > \delta_2 : f(\delta_1) > f(\delta_2)$, where f is the mapping function and δ_1, δ_2 are input distances. The following presents brief descriptions of mappings that we are currently evaluating.

3.7.1 Transformation Method 10

This method performs the $f(\delta) = \delta^{TP}$, where TP is a given power called Transformation Parameter. We are evaluating powers 2, 4 and 6.

3.7.2 Transform Method 8

This mapping assumes a random distribution of distances in a higher dimension, D. Then it performs an analytical derivation of distances in dimension 4. However, since the original distances may not be random, this could end up finding distances in a dimension higher than 4 as well.

3.7.3 Transform Method 9

This starts with transforming distances to 4 dimensions as mentioned in section 3.7.2 and then increases the formal dimension by raising the 4 dimensional distances to power 0.5.

4. CLUSTER VERIFICATION

Even after taking the correct options to produce reliable clusters as elucidated in section 3, we may want to verify the clustering results if possible. One approach that we have been using is to verify the clusters against a given set of consensus sequences. A consensus sequence is usually a biologically determined sequence to represent a group of similar sequences.

Given a M consensus sequences and N sequences we start the pipeline with a $M + N$ sequence set. This results in computing a $(M + N) \times (M + N)$ distance matrix, which becomes the input for the next steps. Once we get both clustering and multi-dimensional scaling results we will visualize the results giving a different color to the M points. If we find good clusters then we will see consensus points appearing within clusters and near the concentration of points in a cluster.

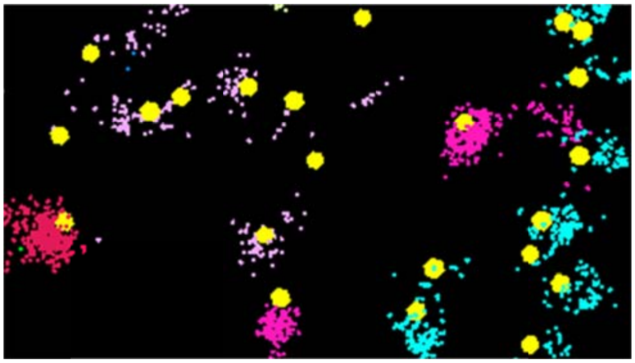


Figure 15. Cluster verification with consensus sequences

The yellow dots in Figure 15 show how consensus sequences appeared within clusters in one of our results. These points lie near the dense region of clusters verifying the accuracy of results.

5. CLUSTER REPRESENTATION

Once we are satisfied with the clustering results we may want to find sequences to represent each cluster. Similar to consensus sequences these would need to be within and near concentrated region of clusters. We name these sequences as cluster centers and present the methods of computing them.

5.1 Sequence Mean

Given a cluster this is the sequence that has the minimum mean distance to other points in the same cluster. So if P is the set of points in a cluster we will find the sequence, $i \in P$, that minimizes $\left(\sum_{j=1}^{C} \delta_{ij}\right)/C$, where $j \in P$ and C is cardinality of P.

5.2 Euclidean Mean

Similar to sequence mean, this method also finds the sequence that has a minimum mean distance to other points in a cluster. However, distances are taken from the three dimensional Euclidean space rather than from distances computed from alignments.

5.3 Centroid of Cluster

This method finds the centroid point of the cluster in the Euclidean space. Then, in the same space, it finds the point nearest to the centroid. The sequence represented by this point is taken as the center for the particular cluster.

5.4 Alternatives to 5.1 and 5.2

The mean distance in both 5.1 and 5.2 may be replaced with the maximum distance as an alternative. Then they will find the sequence that has the smallest maximum distance to other sequences in a given cluster.

6. Description of Data

Studies presented in this paper were based on two sequence data sets where one contains 16S ribosomal RNA and the other a set of fungi sequences. We denote these as 16S rRNA and Fungi data sets.

- 16S rRNA sequences

This contains a total of 1160946 sequences, which reduces to 684769 unique ones based on the actual sequence string. A random selection of 100000 sequences of them is chosen as sample points for interpolation purposes. Figure 16 shows the histogram of sequence lengths for unique sequences and 100K unique sample sequences.

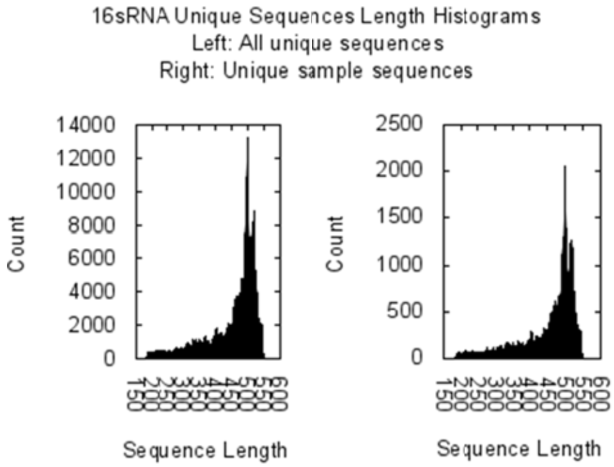


Figure 16. Length histogram of 16S rRNA sequences

The histograms show similar shape indicating an unbiased set of sample sequences with respect to lengths.

- Fungi sequences

This contains a set of fungi sequences received from biologists in Indiana University, Bloomington. It has a total of 957387 sequences where 482158 of them are unique. The biologists were interested in analyzing sequences with lengths greater than 200, which covered a total of 446041 unique sequences. Similar to 16S rRNA sequences, we used a 100000 random sequence set from this as sample sequences.

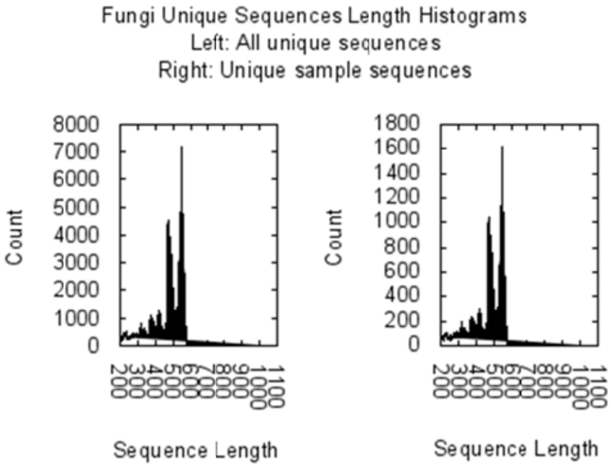


Figure 17. Length histogram of Fungi sequences

7. SUMMARY

In this paper we presented the idea of biological sequence clustering and series of steps involved in the clustering pipeline. We would like to capture and preserve the intrinsic similarities present within the input sequences through the pipeline to obtain reliable clusters in the end. We discussed how visualization could be used to identify ill-defined clusters, the effect of gap penalties, the choice between global and local alignment, different types of distance measures, and distance transformations to reduce dimensionality of input distances to multi-dimensional scaling as measures of ensuring reliability. Next we moved on to the details of verifying clusters when a known set of consensus sequences are available. Visualization aids in this too as one could overlay the consensus sequences over the other sequences in the three dimensional plot to verify the quality of clustering results. Finally, we presented how to find sequences, called centers, to represent each cluster based on intra cluster point distances.

8. REFERENCES

[1] Smith, T. F. and Waterman, M. S. Identification of common molecular subsequences. *J Mol Biol*, 147, 1 (Mar 25 1981), 195-197.

[2] Needleman, S. B. and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48, 3 (Mar 1970), 443-453.

[3] Ekanayake, S. *Heatmaps of Different Distances*. City, 2012.

[4] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. Basic local alignment search tool. *J Mol Biol*, 215, 3 (Oct 5 1990), 403-410.

[5] *The Statistics of Sequence Similarity Scores*. City.

[6] Agrawal, A., Choudhary, A. and Huang, X. Sequence-specific sequence comparison using pairwise statistical significance. *Advances in experimental medicine and biology*, 6962011), 297-306.