Integration of Clustering and Multidimensional Scaling to Determine Phylogenetic Trees as Spherical Phylograms Visualized in 3 Dimensions

Yang Ruan[1], Geoffrey L. House[2], Saliya Ekanayake[1], Ursel Schütte[2,3], James D. Bever[2], Haixu Tang[1,2], Geoffrey Fox[1]
[1]School of Informatics and Computing
[2]Department of Biology
[3]School of Public and Environmental Affairs
Indiana University
Bloomington, Indiana, USA
{yangruan, glhouse, sekanaya, uschuett, jbever, hatang, gcf}@indiana.edu

***Abstract*— Phylogenetic analysis is commonly used to analyze genetic sequence data from fungal communities, while ordination and clustering techniques commonly are used to analyze sequence data from bacterial communities. However, few studies have attempted to link these two independent approaches. In this paper, we propose a method, which we call spherical phylogram (SP), to display the phylogenetic tree within the clustering and visualization result from a pipeline called DACIDR. In comparison with traditional tree display methods, the correlations between the tree and the clustering can be observed directly. In addition, we propose an algorithm called interpolative joining (IJ) to construct and visualize the SP in 3D space. In the experiments, we used the sum of branch lengths to quantify the general fit between the clustering and the phylogenetic tree in SP and Mantel tests to determine how well the same grouping of sequences was preserved between the clustering and the SP. Our results show that DACIDR has a classification accuracy that is similar to a phylogenetic tree generated using a multiple sequence alignment, while having much lower computational cost.**

Keywords—Phylogenetic Tree; Multidimensional Scaling; Microbial Communities; Environmental Genomics

I. INTRODUCTION

The increasing use of high-throughput DNA sequencing techniques to identify microbial communities in the environment has led to a dramatic increase in the size of DNA sequence datasets. The analysis and visualization of these large sequence datasets is a challenge that studies of bacterial diversity and those of fungal diversity have generally approached in different ways. Studies using bacterial DNA sequences typically use clustering approaches such as mothur [1], ESPRIT [2], or UPARSE [3] to group DNA sequences from a sample into operational taxonomic units (OTUs) based on a minimum sequence similarity value (similarity-threshold clustering). The differences between OTUs can also be visualized using multidimensional scaling [4, 5, 6]. In contrast, studies using fungal DNA sequences have typically used phylogenetic analysis in order to identify groups of similar sequences, to visualize the relationships between sequences, and to make inferences about their evolutionary history [7].

However there are important limitations to both similarity-threshold clustering techniques and the phylogenetic analysis techniques. Clustering algorithms that use pairwise sequence alignment (PWA) are computationally faster than creating phylogenetic trees, especially for large numbers of sequences, because they do not require multiple sequence alignment (MSA). Clustering results also allow the clear visualization of extremely large datasets directly. However the clustering results cannot infer the evolutionary relationships between sequences that phylogenetic trees can. Phylogenetic relationships can be important for undescribed taxa that are common among fungi. Also, both methods frequently reduce the size of the dataset being analyzed: for similarity-threshold clustering this reduction is by design through the use of consensus sequences representing each OTU, which are meant to facilitate the visualization and taxonomic identification of the sequences in each cluster; for phylogenetic trees of large datasets, this reduction in the number of sequences is frequently by necessity to allow the computation and clear visualization of the resulting trees.

Here we propose a combined method to address those limitations. For clustering, we use a computationally efficient pipeline called deterministic annealing clustering and interpolative dimension reduction (DACIDR) [8]. Inside DACIDR, a multidimensional scaling (MDS) technique is used to visualize sequence similarity among all sequences in a dataset as a way to infer clusters of similar sequences directly, without the need to define a sequence similarity-threshold (we will refer to this method as MDS cluster visualization). Because MDS cluster visualization allows the observation of sequence similarity of datasets directly, it is a promising technique for determining sequence clusters from high throughput sequencing. However, it is unclear how accurately groups of similar sequences found with the visualization correspond with defined taxonomic groups. In order to evaluate the taxonomic accuracy of groups identified with MDS cluster visualization, a phylogenetic tree was created using maximum likelihood based methods on the same sequence dataset.

As input for MDS cluster visualization and phylogenetic analysis, we used sequences from the variable D2 domain of

the 28S rRNA gene, which is commonly used for taxonomic identification of fungi [9]. All sequences were from species of arbuscular mycorrhizal (AM) fungi because they exhibit a large amount of sequence variation both between species as well as within species [10], which can make them challenging to analyze [11]. The sequence datasets were derived from a combination of: (1) a large-scale AM fungal phylogenetic study [11]; (2) additional sequences obtained from GenBank to increase the taxonomic coverage of the dataset; (3) representative 454 pyrosequences from spores of known AM fungal species that were selected using DACIDR [8]. DACIDR uses pairwise clustering and MDS for robust and scalable sequence clustering and visualization for more than one million sequences [12]. The representative sequences are then selected from each cluster. DACIDR is parallelized to process large datasets on clouds or HPC systems, using MapReduce [13], iterative MapReduce [14] and/or MPI frameworks [15]. A more detailed description of how the sequences are clustered and their biological inference will be presented in later paper.

To compare the consistency between the clustering analysis and the phylogenetic tree, we implemented an algorithm we refer to as interpolative joining (IJ) in order to merge the traditional phylogenetic tree with the MDS cluster visualization into a spherical phylogram (SP). To evaluate how well the SP corresponded to the clustering result from the same dataset, we used a combination of the sum of branch lengths and Mantel tests in our experiments. The different experimental approaches generated similar results that show good agreement between the taxonomic delineations provided by the clustering and those provided by the phylogenetic analysis. This suggests that our proposed clustering technique based on pairwise alignment is a highly suitable alternative to phylogenetic analysis to study microbial communities.

The structure of the paper is organized as follows: Section II discusses existing methods for phylogenetic tree visualization and sequence clustering pipelines; Section III discusses the methods we used for our phylogenetic tree reconstruction, sequence clustering and visualization; Section IV introduces and explains the proposed algorithm for interpolating a phylogenetic tree onto the clustering results; Section V, presents our experimental results and compares our proposed methods to existing tree generation methods; Section VI discusses our conclusions and future work.

II. Related work

There are many different existing clustering algorithms, such as: greedy heuristic methods and hierarchical clustering (both of which are similarity-threshold clustering methods), Bayesian and phylogenetically-aware clustering methods, and MDS cluster visualization demonstrated in this paper.

Greedy heuristic methods define seed sequences to represent the clusters they find and to compare them with all remaining sequences in order to avoid quadratic time complexity. CD-HIT [16] and UCLUST [17] are well-known heuristic clustering methods. They can be very fast to cluster large numbers of sequences, but these algorithms overestimate or underestimate the number of clusters since the similarity threshold is very sensitive with large datasets. Hierarchical clustering also uses a greedy algorithm, but it takes a more structured approach to generating clusters by comparing each additional sequence to all of the sequences already in the cluster [18]. Mothur and ESPRIT are two popular hierarchical clustering methods, but they suffer from quadratic time and space complexity.

Other clustering methods, such as CROP [19], or the phylogenetically-aware GMYC [20] and PTP [21], do not require defined sequence similarity thresholds. Bayesian clustering (CROP) uses a probabilistic approach to define clusters based on the sequence variation that is inherent in the dataset, which also makes it robust to sequencing errors. GMYC uses a maximum likelihood approach to determine the transition point between sequence changes representing speciation events and those representing coalescent events within species [22, 23]. PTP is computationally faster than the GMYC method while also achieving increased clustering accuracy [21]. PTP estimates species clusters using a maximum-likelihood phylogenetic tree produced using the sequences as a guide instead of the coalescent tree, and assumes that each nucleotide substitution has a fixed probability of being the basis for a speciation event [21]. The PTP method is able to give accurate species determinations regardless of the amount of sequence similarity between the species being compared. However both of these methods require either multiple sequence alignment or a guide phylogenetic tree in order to cluster sequences, and therefore are computationally more costly than a clustering algorithm like DACIDR that uses pairwise sequence alignment.

MDS has only been used in cluster visualization in the past few years, but there are many existing algorithms. Newton's method is a simple solution to minimize the STRESS in Eq. (1) and SSTRESS in Eq. (2) [24]. However it uses Hessian to form a basic Newton iteration, and the Hessian construction requires cubic time complexity. A Quasi-Newton [25] method has been proposed to reduce the time complexity of the Newton method to sub-cubic by approximating the Hessian. Multi-Grid MDS [26] has been proposed to solve the isometric embedding problems. The performance was increased dramatically compared to other existing methods because it can be parallelized. The Scaling by Majorizing a Complicated Function (SMACOF) algorithm is one of the MDS algorithms that has been shown to be fast and efficient [27]. Another way of solving the MDS problem is to treat it as a chi-square problem. This can be solved with Manxcat that uses the Levenberg–Marquardt (LMA) [28] algorithm, which is a popular curve fitting function. However, due to the non-linear property of this problem, both of these algorithms could be trapped under local optima. Simulated Annealing and the Genetic Algorithm have been used to avoid the local optima in MDS [17] [18]. However, they suffer from long running times due to their Monte Carlo approach. DA-SMACOF [29] can reduce the time cost and find global optima by using deterministic annealing [30]. But DA-SMACOF assumes all weights are equal to one for all input distance matrices. So we previously added a weighting function to the SMACOF function, called WDA-SMACOF [31]. This uses Conjugate Gradient to avoid the cubic time complexity brought about by weighting and matrix inversion, so that it can converge under $O(N^2)$ time.

The methods used for phylogenetic tree creation have become more standardized compared to clustering techniques. The most commonly accepted methods are probabilistic approaches including maximum likelihood such as RAxML [32] and Bayesian methods such as Mr. Bayes [33]. Because both of these methods incorporate uncertainty into phylogenetic tree construction, they are thought to provide phylogenies that are closely aligned with actual patterns of evolutionary history. Neighbor Joining is a classic method [34], but not as commonly used as the other two methods nowadays.

III. PHYLONENETIC TREE AND CLUSTERING

In this section, we discuss the methods we used to generate the phylogenetic tree as well as the clustering and visualization results. Both of these outputs required sequence alignment beforehand. We did multiple sequence alignment (MSA) for the phylogenetic tree and both MSA and pairwise sequence alignment (PWA) for the clustering. We created the phylogenetic tree using RAxML, and the clustering result was generated using deterministic annealing (DA) based multidimensional scaling (MDS) with the all pair distance matrix. Note that DACIDR was applied on a one million sequence dataset to identify the clusters and their representative sequences used in our experiments. In this paper, we only clustered and visualized the representative sequences with a few hundred other sequences in order to generate the spherical phylogram.

A. Sequence Alignment

As mentioned previously, we were using both MSA and PWA. MSA is used for three or more sequences and it is usually more computationally complex than PWA. It is commonly used in phylogenetic analysis so we chose this method to generate input for RAxML. PWA aims to find an overlapping region of the given two sequences that has the highest similarity as computed by a score measure. The overlap may either be defined over the entire length or over a portion of the two sequences. The former is known as global alignment and latter as local alignment. Needleman-Wunsch (NW) [4] and Smith-Waterman Gotoh (SWG) [35] are two popular algorithms performing these alignments respectively.

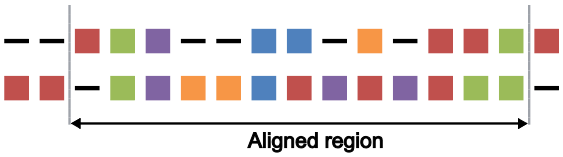


Figure 1 Illustration of Sequence alignment

Figure 1 shows a general sequence alignment with possible end gaps (note a local alignment will not result end gaps). We consider 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. Two parameters governing NW and SWG are the scoring matrix and gap penalties, namely a gap open (GO) and a gap extension (GE) penalty. Alignment algorithms maximize a score measure that is calculated as in Figure 2.

The best alignment algorithm to use may depend on the particular dataset and in certain cases it is possible to obtain alignments that are optimal from the algorithm's point of view, but have little practical value [36].

	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 2 Score of an alignment

B. All Pair Distance Calculation

We align each pair of sequences and compute a distance for each such alignment resulting an all-pairs distance matrix. This serves as the input for remaining algorithms in the DACIDR pipeline. It is possible to define different distance measures [36] for an alignment and we have chosen percent identity (PID) as the distance in this analysis.

Given the alignment between two sequences, let the number of matching pairs in the aligned region be N' and the total number of pairs in the aligned region be N. The PID distance, δ_{PID}, is then computed as given below.

$$\delta_{PID} = 1.0 - \frac{N'}{N} \quad (1)$$

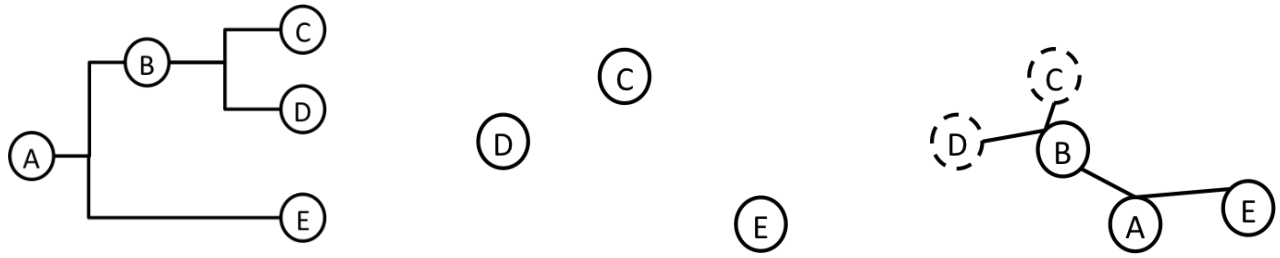
C. Multidimensional Scaling with Deterministic Annealing

MDS is a set of techniques used in dimension reduction. It is used to map original high dimensional data into a target dimension space while preserving the proximity observed in the original dimension space as much as possible. Given a target dimension L, the mapping of points in *L-dimension* can be given by an $N \times L$ matrix X, where each point in the target dimension space is represented as the ith row in X. It is a non-linear optimization problem and the object function that MDS is trying optimize is given as the following:

$$\sigma(X) = \sum_{i<j\leq N} w_{ij}\,(d_{ij}(X) - \delta_{ij})^2 \quad (2)$$

$$\sigma(X) = \sum_{i<j\leq N} w_{ij}\,(d_{ij}^2(X) - \delta_{ij}^2)^2 \quad (3)$$

where w denotes a possible weight, $d_{ij}(X)$ is the Euclidean distance from point i to j in the mapping and δ_{ij} is the original distance from point i to j. This object function is also referred as STRESS or SSTRESS [24]. Note that the original pairwise distance matrix, denoted as Δ must follow three rules: (1) Symmetric: $\delta_{ij} = \delta_{ji}$; (2) Positivity: $\delta_{ij} > 0$; (3) Zero Diagonal: $\delta_{ii} = 0$. We use WDA-SMACOF [31] for our MDS cluster visualization since it can avoid local optima by using DA. DA [30] is an annealing process that finds the global optima of an optimization process instead of local optima by adding a computational temperature to the target object function. By lowering the temperature during the annealing process, the problem space gradually reveals to the original object function. It uses an effective energy function, which is



(a) The cladogram of a tree with 5 nodes

(b) The leaf nodes of the tree in 2D space after dimension reduction

(c) The tree in 2D space after interpolation of the internal nodes

Figure 3 The illustration of a phylogenetic tree in a 2D space

derived through expectation and is deterministically optimized at successively reduced temperatures.

D. Parallezation of the Pipeline

We have improved the efficiency of the parallelization of the pipeline by using a hybrid MapReduce workflow management system [34]. Because all-pair distance calculation is a task-independent application, we used Hadoop [37] for its parallelization inside the workflow. However, it is well-known that Hadoop has a large overhead while running iterative parallel applications, such as MDS applications. Therefore, to avoid that extra computational cost, we use Twister [14], which is an iterative MapReduce framework for parallelization of WDA-SMACOF. The detailed parallelization can be found in [8] and [31]. Finally, since this entire workflow is written in JAVA, it is easy to migrate it to either an HPC cluster or to a Cloud environment.

IV. PHYLOGENETIC TREE DISPLAY WITH CLUSTERING

As mentioned previously, by using DACIDR, each sequence is represented as a point in the target dimension space, i.e. the 3D space. Also, by using RAxML, all the sequences are represented as leaf nodes in the phylogenetic tree. Therefore each leaf node in the phylogenetic tree corresponds to a point in the 3D dimension reduction result. However, traditional tree display software, such as MEGA5 [38] and FigTree [39] only display trees separately from the clustering result, so it is difficult to observe the relationships between the phylogenetic tree and the clustering result.

In this section, we proposed a method, called Interpolative Joining (IJ) to display an existing phylogenetic tree by using the clustered sequences from the same dataset as leaf nodes of the tree. This allows for direct visual comparison between the phylogenetic tree and the sequence clusters. The generated tree can be in either 2D or 3D depending on the target dimension and is referred to either as a circular phylogram in 2D or a spherical phylogram (SP) in 3D. In our study, as our target dimension is 3D, the generated tree will be referred as SP.

A. Distance Calculation

The internal nodes cannot be directly observed because they represent hypothetical ancestor sequences, and therefore the distances from internal nodes to leaf nodes of the generated phylogenetic tree are unknown. By using RAxML, it is possible to calculate distance from an internal node to another node by using the summation over all the branchs between them. For example, in figure 3(a), the distance between point C and E can be calculated by summing over branch(C, B), branch(B, A) and branch(A, E). This distance calculation can generate a pairwise distance matrix for all the nodes based on all the branch lengths. However, the sum of branch lengths does not work to find the distance between pairs of leaf nodes since the pairwise distances between leaf nodes are already known from the MDS cluster visualization results. For example, the distance between leaf node C and D shown in figure 3(b) is clearly not equal to branch(B, C) + branch(B, D). Therefore if the summation over the branches is used for defining distances during interpolation, the result will have a high bias because different distances were used for leaf nodes. Therefore, we chose the distance calculation method used in neighbor joining (NJ) algorithm to calculate the distances between internal nodes based on the existing distances between leaf nodes so that all distances used for visualization are consistent.

The NJ algorithm starts with a completely unresolved tree, whose topology corresponds to that of a star network, and ends once the tree is completely resolved and all branch lengths are known. The core idea of this algorithm is to find a way of constructing a tree that follows the balanced minimum evolution (BME) criterion, which generates the optimal tree topology and minimizes the branch lengths of the tree. Our algorithm IJ used the same strategy to interpolate the phylogenetic tree into the MDS cluster visualization result to generate a SP that will have a minimum total branch length. Nevertheless, if the SP matches the original phylogenetic tree better, the sum of all the branches will be shorter.

The distance calculation used in IJ is similar to the one used NJ, and it can be formulated according to the following: suppose we have n existing points, denoted as $P = \{p_1, p_2, p_3, \dots, p_n\}$. And a point p_i can be represented as a vector $[x_{i1}, x_{i2}, \dots, x_{iL}]$ in *L-dimensions*. The distance between two points p_i and p_j is denoted as $d(p_i, p_j)$ and can be calculated as Euclidean distance using the following equation:

$$d(p_i, p_j) = \sqrt{\sum_{l=1}^{L} (x_{il} - x_{jl})^2} \qquad (4)$$

Given any two points $p_i, p_j \in P$, there are two corresponding leaf nodes in the phylogenetic tree. Their parent is denoted as a new point $\hat{p}$ that can be interpolated into the

target dimension space. The distance from $\hat{p}$ to p_i and p_j can be given in the following equations:

$$d(\hat{p}, p_i) = \frac{1}{2} d(p_i, p_j) + \frac{1}{n-2} \sum_{k=1}^{n} (d(p_i, p_k) - d(p_j, p_k)) \quad (5)$$

Because all of the distances follow three basic rules for Δ mentioned in Section III(C), all distances are symmetric, i.e. $d(p_i, p_j) = d(p_j, p_i)$, and $d(\hat{p}, p_i)$ can be calculated as

$$d(\hat{p}, p_j) = d(p_i, p_j) - d(\hat{p}, p_i) \quad (6)$$

The distances from $p_$to all other points, except p_i and p_j, can be obtained using the following equation where $1 \leq k \leq n$ where $k \neq i$ and $k \neq j$:

$$d(\hat{p}, p_k) = \frac{1}{2}(d(p_i, p_k) + d(p_j, p_k) - d(p_i, p_j)) \quad (7)$$

Note that equation (4) is the Euclidean distance calculation and equation (5) to equation (7) are the calculation of the minimum evolution path for any given two points in *P*, so that for any internal node in the phylogenetic tree, its distance to all other points can be obtained using the equations above.

B. Interpolation

When the distances from the internal nodes to all other points are obtained, we can then interpolate the internal node as a point into the target dimension space. The interpolation was first introduced into the fields of data visualization and clustering to solve the large-scale data problem, also referred to as the *in-sample* and *out-of-sample* problem [40]. First, the original input dataset is split into two parts, one is called the *in-sample* dataset, and the other one is referred to as the *out-of-sample* dataset. Then a clustering or dimension reduction algorithm with a high accuracy can be applied on the *in-sample* dataset to generate the *in-sample* result. Based on the *in-sample* result, an interpolation algorithm with lower time and space cost can be used to generate the result from the *out-of-sample* dataset. The tradeoff of this method is that the interpolation algorithm usually has a lower accuracy then the algorithm applied on the *in-sample* dataset.

In our case, the points in the 3D space that correspond to the phylogenetic tree's leaf nodes are the *in-sample* data, denoted as *P*, and the points representing internal nodes are the *out-of-sample* data, denoted as $\bar{P}$. By using equations (4) to (7), the distance of an *out-of-sample* point $\hat{p}$ to all other *in-sample* points is calculated as the original distance for interpolation, which is denoted as $\hat{\Delta}$. After $\hat{p}$ is interpolated to *L-dimension*, it can be represented as a vector $\hat{x}$ with length *L*. Nevertheless, the *in-sample* points and *out-of-sample* points in the *L-dimension* can be defined as $X = \{X_1, X_2\}$, where $X_1 = \{x_1, x_2, x_3, \dots, x_N\}$ and $X_2 = \{x_{N+1}\}$.

The distance from $\hat{p}$ to all other points can be obtained using equation (1), which is the Euclidean distance in 3D space, denoted as *d(X)*. So for each *out-of-sample* point $\hat{p}$, there is a difference between the Euclidean distance in the *L-dimension* and the original distance, and the object function is given by the following:

$$\sigma(X) = \sum_{i \leq N} w_{i\hat{x}} \, (d_{i\hat{x}}(X) - \delta_{i\hat{x}})^2 \quad (8)$$

The goal of interpolation is to minimize the STRESS value for each of the given *out-of-sample* points so that each *out-of-sample* point can be interpolated to a place where the original distance differs least compared to the *L-dimension* distance. WDA-MI-MDS is a robust iterative algorithm that can interpolate *out-of-sample* points into the target dimension space one by one [31]. For every *out-of-sample* point, the algorithm finds a majorizing function for equation (8), and by using the estimated value of $\hat{x}$ in the previous iteration, it can guarantee a non-increasing STRESS value for $\hat{p}$ as the number of iterations increases. Additionally, it can avoid possible local optima for the STRESS function by using DA. The detailed equations for this algorithm can be found in [31].

Algorithm 1 Interpolative Joining algorithm

Input: P, $\bar{P}$, T, $\bar{T}$
For each pair of siblings (t_i, t_j) in T
  Find their parent $\hat{t}$ in $\bar{T}$
  Find point p_i and p_j in P
  For other point p_k in P
    Compute $d(p_k, p_i)$, $d(p_k, p_j)$ using (4)
  End for
  Compute $d(\hat{p}, p_i)$and $d(\hat{p}, p_j)$ using (5) and (6)
  For other point p_k in P
    Compute $d(\hat{p}, p_k)$ using (7)
  End for
  Use (8) as object function and WDA-MI-MDS to compute $\hat{p}$
  Remove t_i and t_j from T
  Add $\hat{t}$ into T and remove $\hat{t}$ from $\bar{T}$
  Add $\hat{p}$into P and remove $\hat{p}$ from $\bar{P}$,
End for
Return P

C. Tree Generation

Equation (4) and equation (7) give the distance calculation formulas for the internal nodes, which are also referred to as the *out-of-sample* points in previous section, and equation (8) gives the STRESS value of using interpolation for the internal nodes. For each internal node, WDA-MI-MDS can be applied to find its location in the target dimension space. However, not all internal nodes from the phylogenetic tree were selected only based on the leaf nodes. Since in traditional *out-of-sample* problems, the *in-sample* dataset remains the same during interpolation, it is not applicable to use those kinds of algorithms for internal node interpolation. Figure 3(c) gives an example of how the internal nodes are interpolated during neighbor joining. Node A is interpolated based on node E and node B, which is also an internal node for the entire phylogenetic tree shown in Figure 3(a).

To solve that problem, we proposed an algorithm called Interpolative Joining (IJ). In IJ, the *in-sample* dataset needs to

be modified during the interpolation process. Because the *out-of-sample* points are interpolated one by one, each *out-of-sample* point that is already interpolated is added into the *in-sample* dataset and will be considered as an *in-sample* point for subsequent *out-of-sample* points. To do this, the IJ algorithm searches the tree from the bottom up. Every time two leaf nodes are found that share the same parent, those two leaf nodes are used to calculate the coordinates for the internal node. The two leaf nodes will then be removed from the tree, and the newly interpolated internal node will be considered a new leaf node. This is demonstrated in figure 3(c), point C and D are discarded from the leaf node set once node B is interpolated. However, these two *in-sample* points, which correspond to the two leaf nodes, will remain in the *in-sample* dataset. Therefore, the total number of nodes for the input phylogenetic tree will be decreasing and the size of the *in-sample* dataset will be increasing during the interpolation process.

In formal definition, P and $\bar{P}$ are used in terms of *in-sample* and *out-of-sample* points in *L-dimension*; T is the set of leaf nodes and $\bar{T}$ is the set of internal nodes from the phylogenetic tree. Therefore p_i is the representation of t_i in the target dimension space. For each pair of leaf nodes t_i and t_j that have the same parent $\hat{t}$, there is a pair of *in-sample* points which are denoted as point p_i and p_j in P that represents them. Immediately after $\hat{t}$ is found, the $\hat{p}$ that represents it is initialized as a random point and added into $\bar{P}$. After $\hat{t}$ is interpolated into the *L-dimension* space, $\hat{p}$ is removed from $\bar{P}$ and added into P. The t_i and t_j will be removed from T, and $\hat{t}$ is added into T and removed from $\bar{T}$. Nevertheless, $\bar{P}$ will always contain only one *out-of-sample* point during each iteration, where the iteration number equals the number of internal nodes in $\bar{T}$ at the beginning. The detailed process of IJ is illustrated in Algorithm 1. As the calculation of Euclidean distance and WDA-MI-MDS is very fast, generating the SP with a predefined phylogenetic tree and MDS cluster visualization result only takes a few seconds on a single core.

V. EXPERIMENTS

The experiments were carried out on BigRed II, which is a hybrid cluster with a total of 344 CPU nodes with 32 cores per node, and Quarry with a total of 2644 cores with 8 cores per node at Indiana University to process the data with the help of Twister and Hadoop. The clustering and visualization of the sequence datasets were completed using DACIDR. We created a maximum likelihood unrooted phylogenetic tree from the multiple sequence alignment (MSA) with RAxML (Stamatakis 2006) using 100 iterations with the general time reversible (GTR) nucleotide substitution model and with gamma rate heterogeneity (GTRGAMMA). We then used the tree to guide the generation a pairwise distance matrix between all sequences in each of the two MSA datasets using RAxML. These pairwise distance matrices were then used as the reference when testing for the effect of alignment technique and sequence length on consistency between the clustering and the phylogeny. Finally, the IJ was run on a local machine to generate a spherical phylogram (SP), which can be displayed using a data visualization software called PlotViz3 [41].

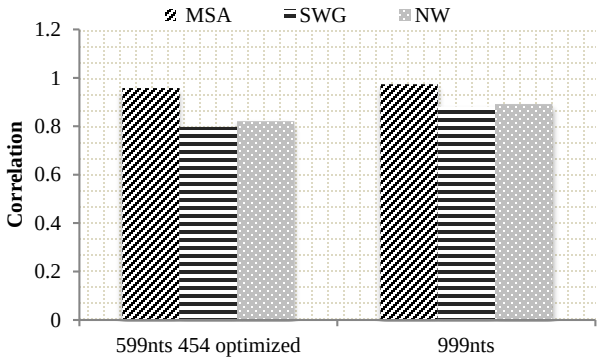


Figure 4 The comparison using Mantel between distances generated by three sequence alignment methods and RAxML

A. *Obtaining sequences*

We first downloaded the sequence alignment of AM fungal sequences from a recent large-scale phylogeny of AM fungi (Krüger et al. 2012) and only retained sequences that contained at least a portion of the 28S rRNA gene. We then collected two sets of additional AM fungal sequences: (1) sequences from GenBank that had confident species attribution in order to supplement the species coverage within the sequence dataset; (2) representative sequences for known AM fungal species obtained from spores using 454 sequencing (Roche, Indianapolis, IN) of the variable and phylogenetically informative D2 domain of the 28S rRNA gene. We applied DACDIR on this dataset to find 126 clusters and then picked a representative sequence for each cluster as part of the dataset. The additional sequences from GenBank were added to the original sequence alignment from [11] using MAFFT [42]. In order to evaluate how different sequence lengths affected the correspondence between phylogenetic trees and clustering, we then created two datasets with sequences that shared the same starting location on the 28S rRNA gene: one dataset contained longer sequences, and the other contained shorter sequences. We first trimmed the MSA and only retained the unique sequences that spanned an extended region beyond the D2 domain (dataset 1, roughly 675 bases long without gaps); then from that subset we retained only the unique sequences that spanned the 454 sequencing start site and the average end position of the 454 sequences (roughly 425 bases long without gaps). Finally, we added the representative 454 sequences to this trimmed alignment using MAFFT as described above to create dataset 2. This gave a MSA for dataset 1 (999nts) with: 801 sequences from [11] and 505 sequences from GenBank for a total of 1306 sequences, and for dataset 2 (599nts with 454 optimized) with: 514 sequences from [11], 380 sequences from GenBank, and 126 representative 454 sequences for a total of 1020 sequences. For this phylogenetic comparison test we selected a smaller set of sequences that still represents the expected range of genetic variability within AM fungi. The RaXML took about 4 hours to finish on the first dataset and 7 hours to finish on the second dataset using 8 cores. And the MDS only took a few minutes to finish on the same dataset using same amount of cores.

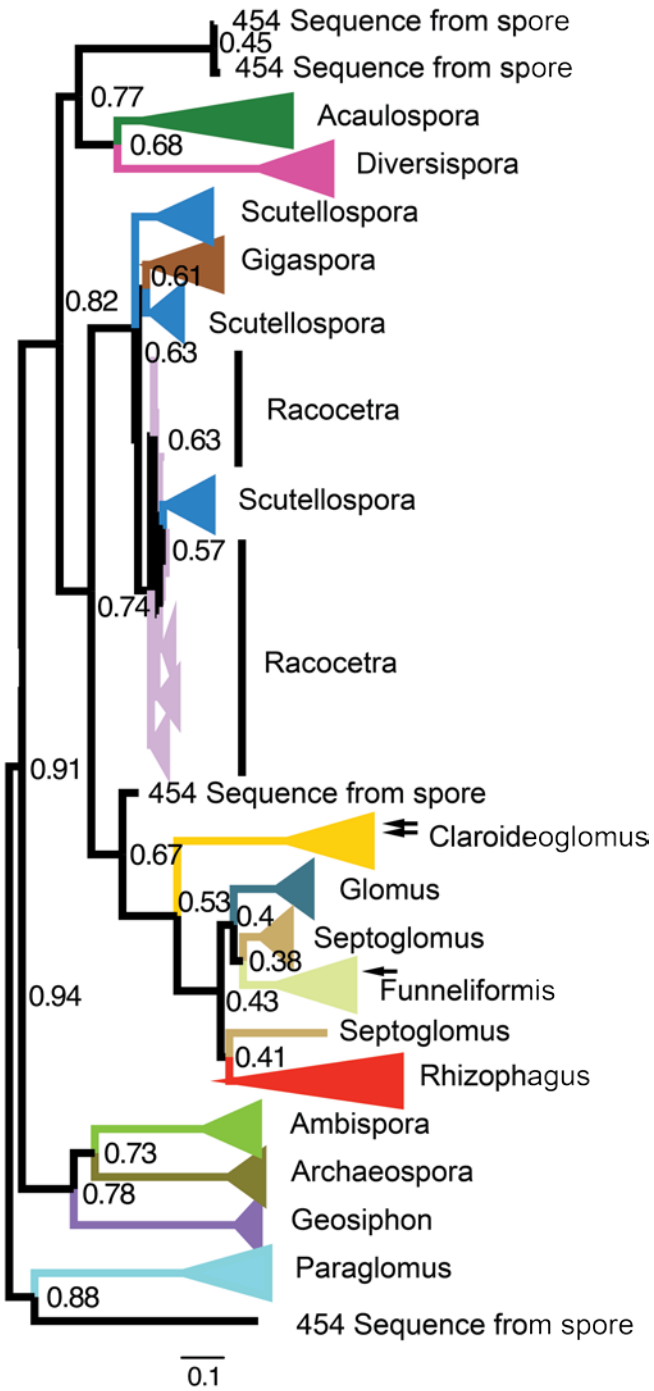
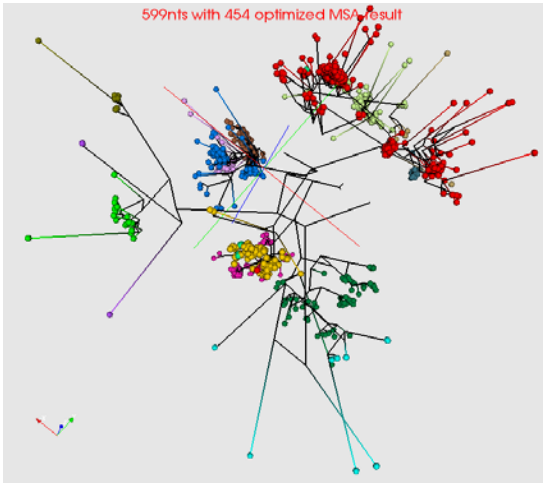
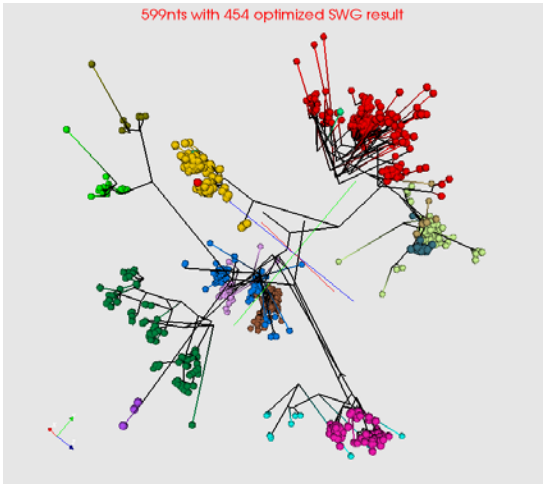


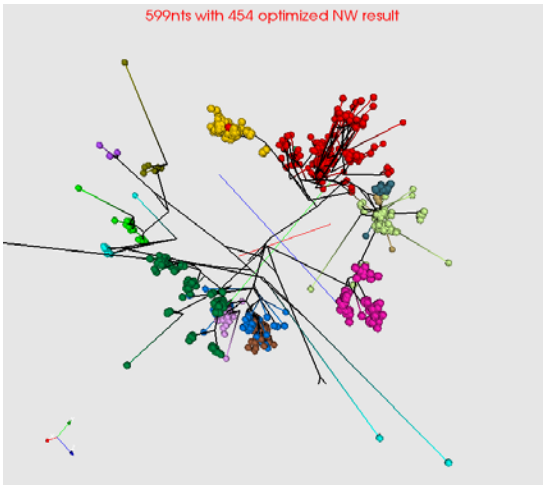
Figure 5 Maximum likelihood phylogenetic tree from dataset 2 that is collapsed into clades at the genus level as denoted by colored triangles at the end of the branches. Branch lengths denote levels of sequence divergence between genera and nodes are labeled with bootstrap confidence values. 454 sequences from spores that are not part of another clade are denoted with the label '454 sequence from spore'. Two sequences in the *Claroideoglomus* clade are instead attributed to *Rhizophagus*, and one sequence in the *Funneliformis* clade is instead attributed to *Septoglomus* (denoted by arrows at the blunt end of the colored triangles). This figure is generated by FigTree.



(a) Multiple sequence alignment (MSA) result



(b) Smith-Waterman pairwise alignment (SWG) result



(c) Needleman-Wunsch pairwise alignment (NW) result

Figure 6 The screenshots of spherical phylogram for using the phylogenetic tree shown in Figure 5 with three different sequence alignments. The colors of the branches in these figures are as same as the colors of the branches shown in Figure 5.

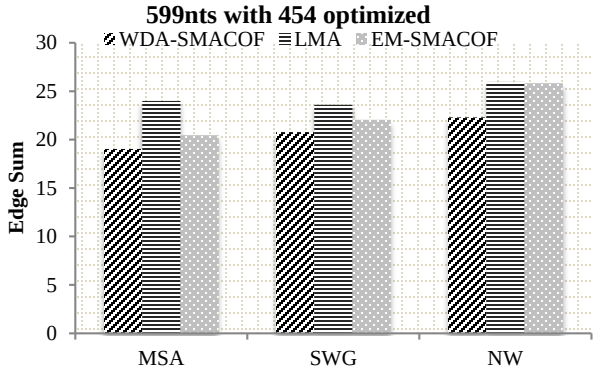
B. Sequence Alignment Comparison

We used the Mantel test in order to evaluate whether pairs of experimental treatments retained the same structure of sequence differences between them.

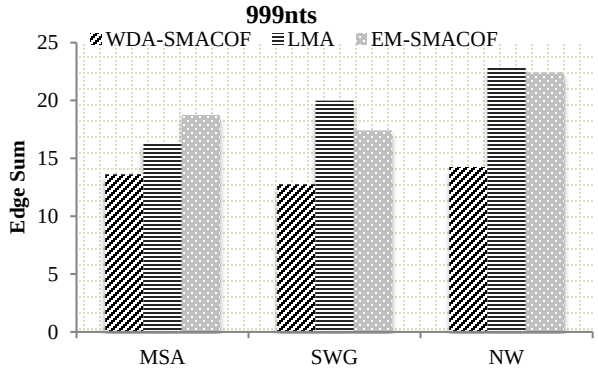
Mantel tests determine whether a correlation between the entries contained in two different pairwise distance matrices is statistically significant by permuting the distance matrices to obtain an empirical *p*-value for the correlation. The treatments consisted of different alignment techniques applied to each of the two different length datasets; comparisons were then made to the RAxML distance matrix from the same dataset. The Mantel tests were performed using the *vegan* package in R (version 3.0.2, R Core Team 2013), and none of the tests had *p*-values greater than 0.001, suggesting all of the measured correlations were likely significant despite the increased type I error (false-positive) rate that can occur with Mantel tests [43].

Figure 4 illustrates the result of the Mantel test applied on MSA and the Pairwise Sequence Alignment (PWA) which includes both the Smith Waterman Gotoh (SWG) and Needle-Wunsch (NW). Using longer sequences (dataset 1) consistently resulted in higher correlations between the reference distance matrix and either of the pairwise alignment techniques. However, both the SWG and the NW pairwise alignment methods gave comparable correlation values for dataset 1 and for shorter sequences (dataset 2). The very high correlations between the RAxML reference matrix and the MSA distance matrix used for MDS cluster visualization regardless of sequence length are expected because the input alignment is identical for both matrices and only the distance calculation method is different. Using pairwise alignments for the same datasets resulted in lower correlations with the RAxML reference matrix, although they still provided a reasonably good fit.

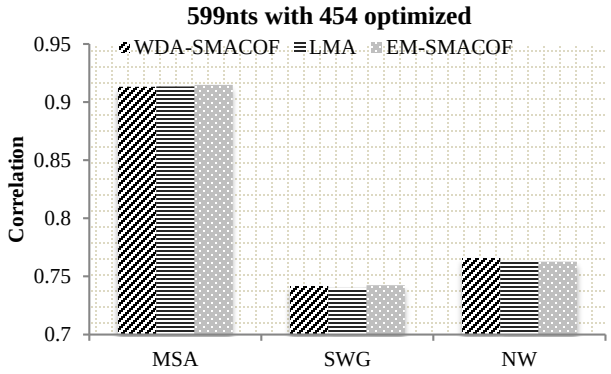
The relationships between genera of AM fungi from the phylogenetic tree created with dataset 2 (Figure 5), was consistent with the current understanding of AM fungal phylogenetic relationships [11], with the exception of *Racocetra*, *Scutellospora*, and *Gigaspora* all being assigned to the same evolutionary group. By comparing the phylogenetic tree (Figure 5) and the SPs (Figure 6), it is possible to visualize the how the branches of the tree correlate with the sequences after MDS. If long branches are required in the interpolated tree in order to connect points that are the same color in the MDS visualization, then the tree does not match well with the MDS result. This is because the sequences on the same branch of the phylogenetic tree are more similar to each other than to other sequences in the dataset, and therefore they should be



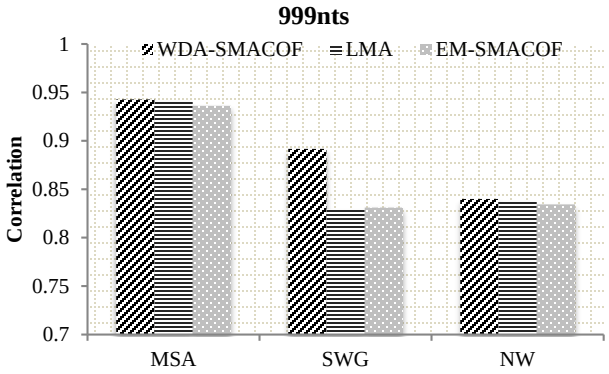
(a) Sum of branch lengths of the SP generated in 3D space on 599nts dataset optimized with 454 sequences



(b) Sum of branch lengths of the SP generated in 3D space on 999nts dataset



(c) Correlation between the distances generated in 3D space and RAxML on 599nts dataset optimized with 454 sequences from Mantel test



(d) Correlation between the distances generated in 3D space and RAxML on 999nts dataset from Mantel test

Figure 7 The sum of branch lengths and Mantel comparison of three different MDS methods using distance input generated from three different types of sequence alignments on two dataset

located close to each other in the MDS visualization as well. The SPs (Figure 6) show that the points with the same color (as color-coded from the genera in the phylogenetic tree, Figure 5), generally group together. There are a few points in the SPs using pairwise alignments (Figure 5(b) and Figure 5(c)) that have longer branches than the points from the SP using the MSA. This is consistent with the fact that the SP generated from the MSA has a better correlation with the phylogenetic tree than the SPs generated from the pairwise alignments (Figure 7(c) and Figure 7(d)). However, the SPs also verify that using pairwise alignments for the MDS generally gives a good fit with the interpolated phylogenetic tree.

C. MDS method comparison

The different methods of MDS affected how well the phylogenetic tree projected using IJ matched the sequences in 3D. WDA-SMACOF is a robust MDS method that can reliably find the global optima, whereas EM-SMACOF can be easily trapped under local optima. The LMA usually had a result that was very similar to EM-SMACOF (Figure 7). The normalized STRESS value for each different input using the different methods was from 0.021 to 0.023, which suggests the distances after dimension reduction have a high similarity to the original distances, and therefore sequence differences were preserved well during MDS; WDA-SMACOF always had the lowest STRESS value compared to the other two methods.

We used the summation over all the branch lengths of the phylogenetic tree in the SP and correlations from the Mantel test to evaluate the differences between these three dimension reduction methods. As mentioned before, the points of the dimensional reduction that connect to the same branches of the SP should be shorter if they match the tree better, which will result in a lower sum of branch lengths. WDA-SMACOF had a much lower sum of branch lengths compared to both LMA and EM-SMACOF (Figure 7 (a) and (b)). This is because the clusters naturally appeared when the STRESS value became lower, but LMA and EM-SMACOF were trapped under the local optima, so there are some points from very small branches of the tree could still be far away from each other in the 3D space and not clustered. In contrast, WDA-SMACOF can reliably find the global optima so that these points from very small branches are always converged into clusters. This is why there were not any excessively long branches for the SP plots generated by using WDA-SMACOF (Figure 6). From the Mantel test correlations (Figure 7 (c) and (d)), although WDA-SMACOF performs better than the other two methods, it shows very little difference between the three.

From Figure 7, the sum over branch lengths is a more sensitive measurement than the Mantel test while evaluating the SPs. However, it also has a higher variance than the Mantel test because it was calculated after IJ. On the other hand, the Mantel test is more robust and shows very little differences while comparing the dimension reduction methods. Therefore, we use both the sum of branch lengths and pairwise correlations from the Mantel test to demonstrate that the interpolated phylogenetic trees closely fit the MDS using WDA-SMACOF, even with pairwise alignments.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a method called Interpolative Joining (IJ) that can be used to project existing phylogenetic trees onto a MDS cluster visualization result in order to generate spherical phylogram (SP), which is much more efficient that traditional displaying method.

Unlike traditional clustering methods that require a similarity-threshold, the WDA-SMACOF used by DACIDR for MDS cluster visualization uses the full range of genetic variability contained in a dataset when determining taxonomic groups because it considers each sequence separately, yet it is still computationally fast. This allows a more natural clustering approach given the inherent variability in the sequence dataset than is possible when using similarity-threshold clustering. In addition, because the taxonomic groups delimited by the clusters visualized by MDS matched those from the phylogenetic tree so closely for AM fungi, computationally slower clustering methods such as GMYC or PTP that use phylogenetic relationships to guide cluster generation may not be required for studies of genetically diverse fungi. In addition to that, WDA-SMACOF can robustly find global optima and be scaled for large datasets [31]. Together these characteristics make DACIDR a promising option for determining taxonomic groups from the increasingly large environmental sequence datasets that are generated by high throughput sequencing.

Overall, even with the genetically diverse AM fungal DNA dataset, we found that the clusters identified by WDA-SMACOF in DACIDR using pairwise sequence alignments accurately defined different taxonomic groups in a way that is in close agreement with a phylogenetic tree generated independently from a multiple sequence alignment of the same dataset. Therefore in our future work, the clustering and visualization using DACIDR appears able to replace the traditional phylogenetic method for the taxonomic analysis of large fungal sequence datasets in studies where either: 1) evolutionary relationships are not of primary interest, or 2) the sequences represent taxonomic groups that are poorly defined in existing sequence databases, which is common when obtaining sequences from environmental samples.

For future improvements to this method, instead of just displaying the representative or consensus sequences from each cluster found from the original input dataset, it is possible to display the tree with entire dataset in the 3D space with the help of IJ. Also the interpolation algorithm used in DACIDR could also be improved to help identify the sequences that are poorly defined. Furthermore, it would be interesting to construct the SP using distances that are first calculated in a higher dimensional space, such as 10D, and are then interpolate the tree into 3D space. This could result in a higher accuracy since a higher dimension space could retain more information from original space. The software to generate SP is available on demand.

ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation under FutureGrid Grant No. 0910812. Our thanks to Judy Qiu from School of Informatics and Computing for providing Twister, and system administrators from University Information Technology Services for providing the support for BigRed2. Sequence data was generated with support from NSF and DoD-SERDP.

References

[1] P. D. Schloss, S. L. Westcott, T. Ryabin *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.

[2] Y. Sun, Y. Cai, L. Liu *et al.*, "ESPRIT: estimating species richness using large collections of 16S rRNA pyrosequences," *Nucleic Acids Research*, vol. 37, no. 10, pp. e76, 2009.

[3] R. C. Edgar, "UPARSE: highly accurate OTU sequences from microbial amplicon reads," *Nat Meth*, vol. 10, no. 10, pp. 996-998, 10//print, 2013.

[4] A. Hughes, Y. Ruan, S. Ekanayake *et al.*, "Interpolative multidimensional scaling techniques for the identification of clusters in very large sequence sets," *BMC bioinformatics*, vol. 13, no. Suppl 2, pp. S9, 2012.

[5] L. Stanberry, R. Higdon, W. Haynes *et al.*, "Visualizing the protein sequence universe," *Concurrency and Computation: Practice and Experience*, 2013.

[6] G. Fox, "Robust Scalable Visualized Clustering In Vector And Non Vector Semi-Metric Spaces," *Parallel Processing Letters*, vol. 23, no. 02, 2013.

[7] U. Koljalg, R. H. Nilsson, K. Abarenkov *et al.*, "Towards a unified paradigm for sequence-based identification of fungi," *Mol Ecol*, vol. 22, no. 21, pp. 5271-7, 2013.

[8] Y. Ruan, S. Ekanayake, M. Rho *et al.*, "DACIDR: deterministic annealed clustering with interpolative dimension reduction using a large collection of 16S rRNA sequences," *in Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pp. 329-336, Orlando, Florida, 2012.

[9] J. Dean, and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107-113, 2008.

[10] J. Ekanayake, H. Li, B. Zhang *et al.*, "Twister: a runtime for iterative mapreduce." *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pp. 810-818, 2010.

[11] J. Willcock, A. Lumsdaine, and A. Robison, "Using mpi with c# and the common language infrastructure," *Concurrency and Computation: Practice and Experience*, vol. 17, no. 7 - 8, pp. 895-917, 2005.

[12] Y. Ruan, Z. Guo, Y. Zhou *et al.*, "HYMR: A Hybrid Mapreduce Workflow System." *In Proceedings of the 3rd international workshop on ECMLS*, pp. 39-48, 2012.

[13] C. L. Schoch, K. A. Seifert, S. Huhndorf *et al.*, "Nuclear ribosomal internal transcribed spacer (ITS) region as a universal DNA barcode marker for Fungi," *Proceedings of the National Academy of Sciences*, vol. 109, no. 16, pp. 6241-6246, April 17, 2012.

[14] R. H. Nilsson, E. Kristiansson, M. Ryberg *et al.*, "Intraspecific ITS variability in the kingdom Fungi as expressed in the international sequence databases and its implications for molecular species identification," *Evolutionary bioinformatics online*, vol. 4, pp. 193, 2008.

[15] M. Krüger, C. Krüger, C. Walker *et al.*, "Phylogenetic reference data for systematics and phylotaxonomy of arbuscular mycorrhizal fungi from phylum to species level," *New Phytologist*, vol. 193, no. 4, pp. 970-984, 2012.

[16] 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, July 1, 2006.

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

[18] S. M. Huse, D. M. Welch, H. G. Morrison *et al.*, "Ironing out the wrinkles in the rare biosphere through improved OTU clustering," *Environmental Microbiology*, vol. 12, no. 7, pp. 1889-1898, 2010.

[19] X. Hao, R. Jiang, and T. Chen, "Clustering 16S rRNA for OTU prediction: a method of unsupervised Bayesian clustering," *Bioinformatics*, vol. 27, no. 5, pp. 611-618, 2011.

[20] J. Zhang, P. Kapli, P. Pavlidis *et al.*, "A general species delimitation method with applications to phylogenetic placements," *Bioinformatics*, vol. 29, no. 22, pp. 2869-2876, 2013.

[21] J. Pons, T. G. Barraclough, J. Gomez-Zurita *et al.*, "Sequence-Based Species Delimitation for the DNA Taxonomy of Undescribed Insects," *Systematic Biology*, vol. 55, no. 4, pp. 595-609, 2006.

[22] J. R. Powell, "Accounting for uncertainty in species delineation during the analysis of environmental DNA sequence data," *Methods in Ecology and Evolution*, vol. 3, no. 1, pp. 1-11, 2012.

[23] T. Fujisawa, and T. G. Barraclough, "Delimiting Species Using Single-Locus Data and the Generalized Mixed Yule Coalescent Approach: A Revised Method and Evaluation on Simulated Data Sets," *Systematic Biology*, vol. 62, no. 5, pp. 707-724, 2013.

[24] A. J. Kearsley, R. A. Tapia, and M. W. Trosset, *The solution of the metric STRESS and SSTRESS problems in multidimensional scaling using Newton's method*, DTIC Document, 1995.

[25] C. T. Kelley, *Iterative methods for optimization*: Siam, 1999.

[26] M. M. Bronstein, A. M. Bronstein, R. Kimmel *et al.*, "Multigrid multidimensional scaling," *Numerical linear algebra with applications*, vol. 13, no. 2 - 3, pp. 149-171, 2006.

[27] I. Borg, and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*: Springer, 2005.

[28] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," *Numerical analysis*, pp. 105-116: Springer, 1978.

[29] S.-H. Bae, J. Qiu, and G. C. Fox, "Multidimensional Scaling by Deterministic Annealing with Iterative Majorization Algorithm." pp. 222-229, 2008.

[30] K. Rose, E. Gurewitz, and G. Fox, "A deterministic annealing approach to clustering," *Pattern Recognition Letters*, vol. 11, no. 9, pp. 589-594, 1990.

[31] Y. Ruan, and G. Fox, "A Robust and Scalable Solution for Interpolative Multidimensional Scaling with Weighting." *In eScience 2013 IEEE 9th International Conference*, pp. 61-69, 2013.

[32] A. Stamatakis, "RAxML Version 8: A tool for Phylogenetic Analysis and Post-Analysis of Large Phylogenies," *Bioinformatics*, 2014.

[33] F. Ronquist, M. Teslenko, P. van der Mark *et al.*, "MrBayes 3.2: Efficient Bayesian Phylogenetic Inference and Model Choice Across a Large Model Space," *Systematic Biology*, vol. 61, no. 3, pp. 539-542, May 1, 2012, 2012.

[34] N. Saitou, and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Molecular biology and evolution*, vol. 4, no. 4, pp. 406-425, 1987.

[35] T. F. Smith, and M. S. Waterman, "Identification of common molecular subsequences," *J Mol Biol*, vol. 147, no. 1, pp. 195-7, Mar 25, 1981.

[36] S. Ekanayake, *Study of Biological Sequence Clustering*, Pervasive Technology Institute, Indiana University, Bloomington, 2013.

[37] T. White, *Hadoop: The Definitive Guide: The Definitive Guide*: O'Reilly Media, 2009.

[38] K. Tamura, D. Peterson, N. Peterson *et al.*, "MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods," *Molecular biology and evolution*, vol. 28, no. 10, pp. 2731-2739, 2011.

[39] A. Rambaut, "FigTree, a graphical viewer of phylogenetic trees," *See http://tree. bio. ed. ac. uk/software/figtree*, 2007.

[40] S.-H. Bae, J. Y. Choi, J. Qiu *et al.*, "Dimension reduction and visualization of large high-dimensional data via interpolation." *In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pp. 203-214, 2010.

[41] "PlotViz3: A cross-platform tool for visualizing large and high-dimensional data", *http://salsahpc.indiana.edu/pviz3/*

[42] K. Katoh, and M. C. Frith, "Adding unaligned sequences into an existing alignment using MAFFT and LAST," *Bioinformatics*, vol. 28, no. 23, pp. 3144-3146, December 1, 2012, 2012.

[43] G. Guillot, and F. Rousset, "Dismantling the Mantel tests," *Methods in Ecology and Evolution*, vol. 4, no. 4, pp. 336-344, 2013.