Evaluate Sequence Clustering using Phylogenetic Trees Interpolated into 3D Clustering

Yang Ruan, Saliya Ekanayake, Haixu Tang, Geoffrey Fox

School of Informatics and Computing

Indiana University

Bloomington, Indiana, USA

{yangruan,

Geoffrey House, Ursel Schuette, Jim Bever

Department of Biology

Indiana University

Bloomington, Indiana, USA

line 4: e-mail address if desired

*Abstract*— Phylogenetic Tree generation is commonly used in studies of fungi data. And visualization and clustering has been commonly used to find the group these bacteria sequences into operational taxonomic units (OTUs). Many studies have been done for these two areas separately, but no studies have been found so far to link these two independent studies. In this paper, we proposed a way, called spherical phylogram (SP) to display phylogenetic tree with a clustering and visualization result. Different from traditional tree displaying methods, the correlations between a tree and clustering could be observed directly. An algorithm called Interpolative Joining is also proposed in this paper to construct the SP in either 2D or 3D space for visualization. In the experiments, we used edge sum to quantify the correlation between the clustering and phylogenetic tree in SP. Compared to a traditional method like Mantel, it is much more sensitive to small differences. Given the results we have over the traditional method and our proposed methods, it is proved that our clustering and visualization method is as accurate as a traditional phylogenetic tree generation method like RaXml to classify the fungi sequences with a much lower computational cost.

Keywords—Phylogenetic Tree; Multidimensional Scaling

# Introduction

The increasing use of high-throughput DNA sequencing 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 (Schloss et al. 2009), ESPRIT (Sun et al. 2009), or UPARSE (Edgar 2013) to group DNA sequences from a sample into operational taxonomic units (OTUs) based on sequence similarity. The differences between OTUs can then be visualized using ordination methods such as principal component analysis (reference?) or non-metric multidimensional scaling (Adam et al. 2010). In contrast, studies using fungal DNA sequences have typically used phylogenetic trees to find groups of similar sequences and to visualize the relationships between sequences in the same dataset (Kõljalg et al. 2013).

However there are important limitations to both the clustering and the phylogenetic tree analysis techniques. Clustering algorithms that use heuristic methods and pairwise sequence alignments are computationally faster than creating phylogenetic trees, especially for large numbers of sequences, because they do not require multiple sequence alignment. Clustering results also allow the clear visualization of extremely large datasets directly, thereby avoiding the frequent need in phylogeny-based analysis to first reduce the number of sequences in the dataset in order to facilitate computation and clear visualization of the resulting trees. However the clustering results cannot infer the evolutionary relationships between sequences that phylogenetic trees can.

In order to test the consistency of the two methods in identifying taxonomic groups, here we combine a computationally efficient clustering pipeline called deterministic annealing clustering and interpolative dimension reduction (DACIDR) (Ruan et al) and a phylogenetic tree created using the same sequence dataset. DACIDR was introduced earlier for robust and scalable sequence clustering and visualization on over millions of sequences. It uses pairwise sequence alignment to calculate the dissimilarity between each pair of sequences. Then it uses pairwise clustering and multidimensional scaling to clustered the data. All of these techniques are parallelized to process large data on multiple compute nodes, using MapReduce, iterative MapReduce and/or MPI frameworks. We improved the parallel efficiency of DACIDR by developing a hybrid workflow model to run it under different cluster environment. The sequences can be grouped into a lower dimension space, where each sequence was represented by a point in 2D or 3D. To overcome the issue mentioned earlier, we proposed an algorithm call Interpolative Joining here to generate a phylogenetic tree diagram called spherical Phylogram (SP) from both the clustering result and a traditional phylogenetic tree. And to evaluate the quality of a SP, which represents the correlations between the clustering result and the tree, a new measurement called edge sum was used in our experiments.

As input for this combined method, we used partial sequences of the 28S rRNA gene, which is commonly used for taxonomic identification in fungi (Schoch et al. 2012). The sequence datasets were derived from a combination of: 1) a large-scale AM fungal phylogenetic study (Krüger et al. 2012); 2) additional sequences obtained from GenBank to increase the taxonomic coverage of the dataset; 3) 454 consensus sequences from spores of known AM fungal species. We carried out experiments using the proposed methods are a well-known traditional method called Mantel, and they generate very similar result that there is a very good agreement between the taxonomic delineations provided by the clustering and those provided by the phylogenetic tree.

The structure of the paper is organized as following: Section II discusses existing methods for phylogenetic tree visualization and sequence clustering pipelines; Section III discussed the methods we used for our phylogenetic tree generation, sequence clustering and visualization. Section IV introduces and explains proposed algorithm for phylogenetic tree displaying with clustering result; In Section V, we present our experimental result and compare our proposed methods to existing popular tree generation methods; Followed by our conclusion and future work in Section VI.

# Related work

There are many different algorithms that cluster DNA or RNA sequences of microbes into taxonomic groups. Greedy heuristic methods and hierarchical clustering work by defining sequence similarity thresholds to determine the membership of sequences in each cluster. Other methods including Bayesian clustering and phylogenetically-aware clustering methods are more flexible in delimiting clusters based on the characteristics of the input sequence dataset.

Greedy heuristic methods define a single sequence ‘seed’ from the full dataset to compare with all remaining sequences. For each remaining sequence in the dataset, if the genetic distance between the sequence and the seed is within the specified sequence similarity threshold, then the sequence joins that cluster; otherwise it forms its own cluster and becomes an additional seed for sequence comparison. CD-HIT and UCLUST are very well known heuristic clustering methods. They can be very fast to cluster large scale of data, but these algorithm overestimate the cluster number found in a large dataset..

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 and allowing a choice of three different cluster membership rules. These membership rules specify that clusters only comprise sequences that are within the specified sequence similarity threshold of either: 1) every other sequence in the cluster (complete linkage), or 2) at least one other sequence in the cluster (single linkage), or that the average sequence similarity among of all sequences in a cluster be less than the similarity threshold (average linkage) (Huse et al. 2010). Depending on whether the sequence passes the membership rule for a given cluster, it either joins the cluster or is designated as its own cluster. The process is then repeated with the remaining sequences until the clustering is complete.

A more flexible approach to generating sequence clusters (CROP) uses a Bayesian clustering approach to define clusters based on the sequence variation that is inherent in the dataset (Hao et al. 2011). This method is computationally efficient and its probabilistic approach makes it robust to sequencing errors in the sequences being clustered. It can also have increased clustering accuracy compared to the hierarchical clustering methods mothur and ESPRIT (Hao et al. 2011), but only if the sequence divergence between clusters is large (Zhang et al. 2013), which we would not expect to be the case with AM fungi (Nilsson et al. 2008).

Other clustering methods that use inferences about phylogenetic relationships between sequences also do not require defined sequence similarity thresholds, although these methods are more computationally intensive because they require multiple sequence alignment. The Generalized Mixed Yule Coalescent (GMYC) method (Pons et al. 2006) is the most widely used of these clustering techniques (Zhang et al. 2013). From the pattern of single gene evolution (the coalescent) derived from a given sequence dataset (like for the 28S rRNA gene), GMYC uses a maximum likelihood approach to determine the transition point sequence changes representing speciation (Yule) events to those representing coalescent events (population divergence within the same species) (Powell 2012, Fujisawa and Barraclough 2013).

A recently proposed alternative to the GMYC method is the Poisson Tree Process (PTP), which is computationally faster than the GMYC method while also achieving increased clustering accuracy (Zhang et al. 2013). The PTP estimates species clusters using a maximum-likelihood phylogenetic tree produced from the sequences as a guide (instead of the coalescent tree required for the GMYC method), and assumes that each nucleotide substitution has a fixed probability of being the basis for a speciation event (Zhang et al. 2013). The PTP is able to give accurate species determinations regardless of the amount of sequence similarity between the species being compared. However the PTP still requires either multiple sequence alignment or a guide phylogenetic tree in order to cluster sequences, and therefore is computationally more costly than a clustering algorithm that uses pairwise sequence alignment.

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 (RAxML) and Bayesian methods (Mr. Bayes ). Because both of these methods incorporate uncertainty phylogenetic tree construction, they are thought to provide phylogenies that are closely aligned with actual patterns of evolutionary history.

# Phylonenetic Tree and clustering

In this section, we discuss the methods we used to generate the phylogenetic tree and the clustering and visualization result. Both of these output requires sequence alignment beforehand. We did multiple sequence alignment for phylogenetic tree and pairwise sequence alignment for the clustering. Then the phylogenetic tree was created using RAxML (version 7.4.2, Stamatakis 2006). It took 100 iterations with the general time reversible (GTR) nucleotide substitution model and with gamma rate heterogeneity (GTRGAMMA). The clustering result was generated using multidimensional scaling (MDS) with deterministic annealing (DA) on the all pair distance matrix after pairwise sequence alignment.

## Sequence Alignment

Sequence alignment 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 a portion of the two sequences. The former is known as global alignment and latter as local alignment. Needleman-Wunsch[ref] and Smith-Waterman[ref] 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 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.

 Two parameters governing both these algorithms are scoring matrix and gap penalties, namely gap open and gap extension. Alignment algorithm maximizes a score measure calculated as shown in Figure 2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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  |
|  |

Figure 2. Score of an alignment

## 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 DACIDR pipeline.

It is possible to define different distance measures [Ref qls report] 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 aligned region be and the total number pairs in the aligned region be . The PID distance, , is then computed as given below.

|  |  |
| --- | --- |
|   | (1) |

## Multidimensional Scaling with Deterministic Annealing

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

|  |  |  |
| --- | --- | --- |
|  |   | (2) |
|  |   | (3) |

where *w* denotes a possible weight, is the Euclidean distance from point *i* to *j* in the mapping and is the original distance from point *i* to *j.* This object function is aslo referred as STRESS or SSTRESS. Note that the original pairwise distance matrix, denoted as must follows the rules: (1) Symmetric: . (2) Positivity: . (3) Zero Diagnosal: . Scaling by Majorizing a Complicated Function (SMACOF) algorithm is one of the MDS algorithms that have been proved to be fast and efficient (I Borg). Another way of solving MDS problem solve it as a chisq problem. And Levenberg–Marquardt (LMA) algorithm is a popular curve fitting function to solve the chisq. However, due to the non-linear property of this problem, both of these algorithms could be trapped under local optima. Therefore, we use deterministic annealing (DA) technique to solve this issue.

DA is an annealing process that finds 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. Different from Simulated Annealing, which is based on Metropolis algorithm for atomic simulations, it neither rely on the random sampling process nor random decisions based on current state. DA uses an effective energy function, which is derived through expectation and is deterministically optimized at successively reduced temperatures. Seung-Base has proposed DA-SMACOF for this problem. In addition to that,[15] added a computational temperature and weighting function to the SMACOF function, called WDA-SMACOF. It has been proved to be reliable, fast, and robust.

## Parallezation of Pipeline

We have improved the efficiency of the parallelization of the pipeline by using a hybrid MapReduce workflow management system [ref HyMR]. Because all-pair distance calculation is a task independent application, so we used Hadoop for the parallelization of all-pair distance calculation 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, which is an iterative MapReduce framework for parallelization of WDA-SMACOF. The detailed parallelization can be found in [refe HyMR] and [refer WDA-SMACOF]. Finally, since this entire workflow is written in JAVA, so it is easy to migrate it to either HPC cluster or a Cloud environment.

# 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. And by using RAxML, all the sequences are represented as leaf nodes in the generated phylogenetic tree. So each leaf node in the phylogenetic tree correspond to a point in the 3D dimension reduction result. However, traditional tree displaying techniques mentioned earlier only display trees separately from the clustering result, so it is difficult to observe the relationships between the tree generation result and the clustering result.

|  |  |  |
| --- | --- | --- |
|  (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 |

 In this section, we proposed a method, called Interpolative Joining (IJ) to display phylogenetic tree from clustering result, so that the correlations between evolutionary relationships and gene families among various biological species can be observed directly by viewing the phylogenetic tree. As the phylogenetic tree’s internal nodes, which are hypothetical taxonomic units, cannot be directly observed from the clustering and visualization result, they are interpolated into the 3D space. And the edge of the tree represents the branch from a parent to its child. It is drawn simply using a line to connect the parent and child, which are two points in the target dimension space. The generated tree can be in either 2D or 3D depends on the target dimension. And it is referred either circular phylogram in 2D or spherical phylogram (SP) in 3D. In our study, as we use 3D space as our target dimension, the generated tree will be referred as SP in this paper.

## Distance Calculation

As the internal nodes can not be directly obseved, so the distances from internal nodes to leaft nodes from the generated phylogenetic tree is unknown. And by usng RAxML, it is possible to get distance from an internal node to another node by using the summation over all the edges between them. For example, in figure 3(a), the distance between point C and E can be calculated by summing over edge(C, B), edge(B, A) and edge(A, E). This distance calculation can generate a pairwise distance matrix for all the nodes in the tree once we know all the edge lengths. However, the all pairs distances between leaft nodes are known from the clustering and visualization result. And the distance between leaft node C and D shown in figure 3(b) is clearly not equal to edge(B, C) + edge(B, D). So if the summation over edge is used as distance for interpolation, the result will have a high bias as the distances used for internoal nodes and different from the distances used for leaft nodes. Therefore, we choosed the distance calculation method used in neighbor joing to calculat the distances between internal nodes based on the existing distances between leaf nodes so that all distances used for visualization are consistant.

Neighbor joining (NJ) is a phylogenetic tree generation method proposed by Naruya Saitou and Masatoshi Nei in 1987. The algorithm starts with a completely unresolved tree, whose topology corresponds to that of a star network, and ends where the tree is completely resolved and all branch lengths are known. The core idea of this algorithm is find a way of contructing a tree which follows the balanced minimum evolution (BME) criteron. BME optimal topology is a technique which minimizes the edge lengths of the tree. And our algorithm IJ used the same strategy in generating the phylogenetic tree from clustering result, so that the tree will have a minimum branch lengths for displaying purpose. Nevertheless, if a tree generated from the clustering result matches the exisiting tree better, the sum over the edges will be shorter.

The distance calculation used in IJ is similar to the one used NJ, and it can be formulated as following: Suppose we have *n* existing points, denoted as . And a point can be represented as a vector in the target dimension *L*. The distance between two points and is denoted as and can be calculated as Euclidean distance using following equation:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Given any two points , there are two corresponding leaf nodes in the phylogenetic tree. And their parent is denoted as a new point . can be interpolated into the target dimension space. The distance from to and can be given in following equations:

|  |  |
| --- | --- |
|  | (5) |

Because all of the distances follows three basic rules for pairwise distances. Thus all distances are symmetric, i.e. , and can be calculated as

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

The distances from to all other points, except pi and pj, where where and , can be obtained using the following equation:

|  |  |
| --- | --- |
|  | (7) |

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

## Interpolation

Even if the distances from 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 brought into the data visualization and clustering area to solve the large-scale data problem. And it was also referred as *in-sample* and *out-of-sample* problem. First, the original input dataset is split into two parts, one is called *in-sample* dataset, and the other one is referred *out-of-sample* dataset. Then a clustering or dimension reduction algorithm with a high accuracy could 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 could 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 phylogenetic tree’s leaf nodes corresponding points in the 3D space are the *in-sample* data, denoted as *P*, and the points represent internal nodes are the *out-of-sample* data, denoted as . By using equation (4) to (7), the distance of an *out-of-sample* point to all other *in-sample* points is calculated as the original distance for interpolation, which is denoted as . After is interpolated to *L-dimension*, it can be represented as a vector with length *L*. So the *in-sample* points and *out-of-sample* points in the L-dimension can be defined as , where and .

The distance from 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 , there is a difference between the Euclidean distance in the *L-dimension* and the original distance, and the object function is given as following:

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

The goal of interpolation is to minimize the STRESS value for each of the given *out-of-sample* point. So that this *out-of-sample* point can be interpolated to a place where the original distances differs least to the target dimension distances. WDA-MI-MDS is a robust iterative algorithm which can interpolate *out-of-sample* points into the target dimension space one by one. For every *out-of-sample* point, the algorithm finds a majorizing function for equation (8). And by using the estimated of in previous iteration, it can guarantee a non-increasing STRESS value for as iterations goes up. And by using deterministic annealing technique, it can avoid possible local optima for STRESS function. The detailed equations for this algorithm can be found in [some paper].

## Tree Generation

Equation (4) and equation (7) gives the distance calculation fomulas for the internal nodes, which is also referred as out-of-sample point 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 nodes 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 phylogenec tree shown in Figure 3(a).

**Algorithm 1 Interpolative Joining algorithm**

Input: , , ,

For each pair of siblings (,) in *T*

 Find their parent in

 Find point and in *P*

 For other point in *P*

 Compute , using (4)

 End for

 Compute and using (5) and (65)

 For other point in *P*

 Compute using (7)

 End for

 Use (8) as object function and WDA-MI-MDS to compute

 Remove and from *T*

 Add into *T* and remove from

 Add into *P* and remove from ,

End for

Return P

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, so for each *out-of-sample* point that is already interpolated, it is added into the *in-sample* dataset and will be considered as an *in-sample* point for later *out-of-sample* point. As shown in figure 3(c), point C and D are discard from the leaf node set once node B is interpolated. The IJ algorithm search the tree from bottoms up, every time two leaf nodes shared a same parent is found, And the two leaf nodes which are used to calculate the coordinates for the internal node will be removed from the tree, where the newly interpolated internal node will be considered as a new leaf node. 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.

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

In formal definition, and are used in terms of *in-sample* and *out-of-sample* points in *L-dimension*, and is the set of leaf nodes and is the set of internal nodes from the phylogenetic tree. Therefore is the representation of in the target dimension space. For each pair of leaf nodes and that have the same parent , there is a pair of *in-sample* points which are denoted as point and in *P* that represents them. Immediately after is found, a random represents that is added into , and once it is interpolated into the *L-dimension* space, it is removed from and added into *P*. The and will be removed from *T*, and is added into *T* and removed from . Nevertheless, will always contain only one *out-of-sample* point during each iteration, where the iteration number equals the number of internal nodes in at beginning. The detailed process of IJ is illustrated in Algorithm 1. As Euclidean distance calculation and WDA-MI-MDS executed very fast, to generate a phylogenetic tree plot with clustering in 3D only takes a few seconds on a single core.

# Experiments

The experiments were carried out on BigRed II, which is a hybrid cluster with a total number of 344 CPU nodes and 676 CPU/GPU nodes. Each CPU node contains two AMD Opteron 16-core Abu Dhabi x86\_64 CPUs and 64 GB of RAM. It’s a Cray Linux Environment and we used the cluster compatity mode to process the data with the help of Twister and Hadoop. The clustering and visualization of the 599nts and 999nts data were using DACIDR. We created a maximum likelihood unrooted phylogenetic tree from the multiple sequence alignment with RAxML (Stamatakis 2006) using 100 iterations with the general time reversible (GTR) nucleotide substitution model and with gamma rate heterogeneity (GTRGAMMA). After creating the phylogenetic tree, we used it to guide the generation a pairwise distance matrix between all sequences in each of the two multiple sequence alignment datasets using RAxML. These pairwise distance matrices were then used as the reference when comparing pairwise distance matrices created from: 1) different sequence alignment methods, and 2) different dimensional reduction methods. Finally, the IJ was running on local machine since this algorithm was fast enough to be run on a single core.

## Obtaining sequences

 We first downloaded the sequence alignment of AMF sequences from a recent large-scale phylogeny of AMF (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, and 2) consensus sequences for known AM fungal species obtained from AM fungal spores using 454 sequencing (Roche) of the variable and phylogenetically informative D2 domain of the 28S rRNA gene and then clustered with AbundantOTU (Ye 2010) using a 99% sequence similarity threshold. The additional sequences from GenBank were added to the original sequence alignment from Krüger et al. (2012) using MAFFT (Katoh and Frith 2012). MAFFT allows the insertion of gaps in the original alignment if the new sequences do not fit, and is therefore a more flexible alignment technique compared to some others, such as SINA (Pruesse et al. 2012). In order to evaluate how different sequence lengths affect the correspondence between phylogenetic trees and clustering, we then created two datasets with sequences that shared same starting location on the 28S rRNA gene: one dataset contained longer sequences, and the other contained shorter sequences. We first trimmed the multiple sequence alignment 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 consensus 454 sequences to this trimmed alignment using MAFFT as described above to create dataset 2. This gave a multiple sequence alignment for dataset 1 with: 801 sequences from Krüger (2012) and 505 sequences from GenBank for a total of 1306 sequences, and for dataset 2 with: 514 sequences from Krüger (2012), 380 sequences from GenBank, and 126 consensus 454 sequences for a total of 1020 sequences. We note that DACIDR is capable of efficiently clustering millions of sequences (Ruan et al. 2012), but for this phylogenetic comparison test we selected a smaller set of sequences that still represents the expected range of genetic variability within AM fungi.

|  |  |  |
| --- | --- | --- |
| **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).** |  | **(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 4. The colors of the branches in these figures are as same as the colors of the branches shown in the phylogram in figure 5.** |

## Sequence Alignment Comparison

We used Mantel tests and the spherical Phylogram 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. Although Mantel tests have been shown to have a high type I error (false-positive) rate, especially in cases of autocorrelation between the two distance matrices, there are not currently alternative methods for the type of sequence data considered here (Guillot and Rousset 2013). 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 any increased type I error rate.

|  |  |
| --- | --- |
| **(a) Edge sum of the SP generated in 3D space on** **599nts dataset optimized with 454 sequences** | **(b) Edge sum 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 edge sum and Mantel comparison of three different MDS methods using distance input** **generated from three different types of sequence alignments on two dataset** |

Figure 4 illustrates the result of the Mantel test applied on Multiple Sequence Alignment (MSA) and the Pairwise Sequence Alignment(PWA) which includes Smith Waterman Gotoh (SWG) and Needle-Wunsch(NW). It shows that 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 clustering MSA distance matrix regardless of sequence length are expected because the input alignment is identical for both matrices and only the distance calculation method is different.

The spherical Phylogram is given in figure 6 and the phylogenetic tree used in generating the plots are given in figure 5. From the plots we can directly observe the how the tree correlates with the plot. If there are some long branches over the entire space with the same color of points, it means that the tree does match well with the clustering. This is because the sequences under the same branch should be near each other in a clustering result. Figure 6 shows that all of the points within the same color matches well with the tree in figure 5. There are a few points in figure 5(b) and figure 5(c) has a longer branch than the points in MSA, which suggests that the SP generated from MSA has a better correlation with the tree than from pairwise alignment, which is consistent with the result from Mantel test. Most importantly, these plots proved that PWA and clustering has a very similar result with MSA and clustering, and it matches well with the result generated from MSA and RAxML.

## MDS method comparison

During the experiment, we found out that different method of MDS can show various affect on the phylogenetic tree displaying in 3D using IJ. WDA-SMACOF is a robust method that can always find global optima, where EM-SMACOF can be easily trapped under local optima. The LMA usually has a result which is very similar to EM-SMACOF. The normalized STRESS value for each different input using different methods is around 0.021 to 0.023, which suggests the distances after dimension reduction has a high consistency from the original distances.

We use edge sum over all the branches of the phylogenetic tree in the spherical Phylogram and correlations from the Mantel test to evaluate the differences between these three dimension reduction methods. As mentioned before, the points under the same branches of the tree should be shorter if the spherical Phylogram matches the tree better. And spherical Phylogram was generated using Interpolative Joining based on the dimension reduction (clustering) result. Therefore, the edge sum will be lower if the clustering result is better. From Figure 7 (a) and (b), we can observe that WDA-SMACOF has a much lower edge sum compared to both LMA and EM-SMACOF. This is because when LMA and EM-SMACOF trapped under local optima, there are some points from a very small branch of the tree could still be far away from each other and not clustered. But since WDA-SMACOF can find global optima, these very small clusters are always converged. Therefore, there were not any long branches for the plot generated by using WDA-SMACOF. The plots from figure 6 are all generated using WDA-SMACOF, which also confirms the result. And Figure 7 (c) and (d) are the result from Mantel, where the pairwise distance matrix are calculated using the points after dimension reduction, and compared with the distance matrix generated by RAxML. And although WDA-SMACOF performs better than the other two methods, it shows very little difference. That is the reason we are using edge sum, which is a much more sensitive method to evaluate the results as well. The drawback of edge sum is that it is not as robust as Mantel because of the possible variation brought IJ algorithm. So in our experiment, we use both the edge sum and Mantel and measurements to confirm our results with the visualization of phylogenetic tree as spherical Phylogram.

From the other aspect, the significant correlations from Mantel indicate that all three retain the distance relationships among the sequences that were present in the original dataset. Because the reference distance matrix for each of these comparisons was generated using the phylogenetic tree, these significant correlations also demonstrate that clusters generated by the MDS closely match the toplogy of the phylogenetic tree. There were apparent differences in correlations between the MSA and PWA alignments after the clustering and dimensional reduction. However, the low stress values relative to the differences in the MSA and the PWA alignment correlations implies that most of those differences occurred during the sequence alignment itself and not during the clustering process or the dimensional reduction.

# ConclusionS and Future work

On one side, we proved that the clustering can replace the traditional phylogenetic tree 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. One the other side, IJ can be used to draw diagrams of phylogenetic trees with clustering result, which is much more efficient than the traditional displaying methods.

Overall, even with the genetically diverse AM fungal DNA dataset, the DACIDR clustering algorithm using pairwise sequence alignments accurately delimited 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. Indeed, the clustering was even able to resolve species-level differences in the genera Acaulospora, Ambispora, Diversispora, and Paraglomus.

In our phylogenetic tree created with the sequence dataset of the D2 domain from the 28S rRNA gene, the genera Scutellospora, Racocetra, and Gigaspora formed a clade, which is consistent with the current understanding of AM fungal phylogenetic relationships (Redecker et al. 2013), but they all shared the same most recent common ancestor, and so were paraphyletic relative to each other. This was also evident in the clustering where there was little difference between the three genera, and especially between Scutellospora and Racocetra.

Like any clustering method, DACIDR by itself is not able to reconstruct the evolutionary relationships between taxa that is possible with a phylogenetic tree. However, we demonstrate that it can provide a quick and accurate determination of fungal taxonomic groups within a large sequence dataset. DACIDR can also make use of the full range of genetic variability when determining taxonomic groups from these large datasets because it can analyze all of the sequences in the dataset simultaneously. In contrast, phylogenetic techniques can only analyze smaller numbers of sequences and therefore they require that large datasets first be reduced in size, usually by the creation of similarity-based consensus sequences, which remove much of the original genetic variability present in the sequence dataset. For this reason, DACIDR may represent an effective way to determine taxonomic groups within the increasingly common large sequence datasets that are created by applying high throughput sequencing to environmental samples.

##### Acknowledgment

##### References

1. A. Hughes, Y. Ruan, S. Ekanayake, S.-H. Bae, Q. Dong, et al. (2012). "Interpolative multidimensional scaling techniques for the identification of clusters in very large sequence sets." *BMC Bioinformatics* 13(Suppl 2): S9.
2. L. Stanberry, R. Higdon, W. Haynes, N. Kolker, W. Broomall, et al. "Visualizing the Protein Sequence Universe." *Proceedings of ECMLS’12 of ACM HPDC 2012*, Delft, Netherlands, ACM, 2012.
3. Y. Ruan, S. Ekanayake, M. Rho, H. Tang, S.-H. Bae, 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. ACM, 2012.
4. S.-H. Bae, J. Y. Choi, J. Qiu, and G. C. Fox. "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. ACM, 2010.
5. K. Rose, E. Gurewitz, and G. C. Fox, “A deterministic annealing approach to clustering,” *Pattern Recognition Letters*, vol. 11, no. 9, pp. 589–594, 1990.
6. Kearsley A, Tapia R, Trosset M. "The solution of the metric STRESS and SSTRESS problems in multidimensional scaling using Newton’s method." *Computational Statistics*, 13(3):369–396, 1998.
7. I. Borg and P. J. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. New York, NY, U.S.A.: Springer, 2005.
8. Bae, Seung-Hee, Judy Qiu, and Geoffrey C. Fox. "Multidimensional Scaling by Deterministic Annealing with Iterative Majorization algorithm." In *e-Science (e-Science), 2010 IEEE Sixth International Conference on*, pp. 222-229. IEEE, 2010.
9. M. Brusco, “A simulated annealing heuristic for unidimensional and multidimensional (city-block) scaling of symmetric proximity matrices,” *Journal of Classification*, vol. 18, no. 1, pp. 3–33, 2001.
10. M. W. Trosset and C. E. Priebe. "The out-of-sample problem for classical multidimensional scaling." *Computational Statistics and Data Analysis*, 52(10):4635–4642, 2008.
11. J.Ekanayake, H. Li, B. Zhang, T. Gunarathne, et al. "Twister: A Runtime for iterative MapReduce." *Proceedings of MapReduce’10 of ACM HPDC 2010*, pp. 810-818. ACM, 2010.
12. Y. Ruan, Z. Guo, Y. Zhou, J. Qiu, and G.C. Fox. "HyMR: a Hybrid MapReduce Workflow System." *Proceedings of ECMLS’12 of ACM HPDC 2012*, Delft, Netherlands, ACM, 2012.
13. PlotViz - A tool for visualizing large and high-dimensional data. <http://salsahpc.indiana.edu/pviz3/>
14. Edgar, R. C. 2013. UPARSE: highly accurate OTU sequences from microbial amplicon reads. Nat Meth 10:996-998.
15. Katoh, K. and M. C. Frith. 2012. Adding unaligned sequences into an existing alignment using MAFFT and LAST. Bioinformatics 28:3144-3146.
16. Kõljalg, U., R. H. Nilsson, K. Abarenkov, L. Tedersoo, A. F. S. Taylor, et al. 2013. Towards a unified paradigm for sequence-based identification of fungi. Molecular Ecology 22:5271-5277.
17. Krüger, M., C. Krüger, C. Walker, H. Stockinger, and A. Schüßler. 2012. Phylogenetic reference data for systematics and phylotaxonomy of arbuscular mycorrhizal fungi from phylum to species level. New Phytologist 193:970-984.
18. Pruesse, E., J. Peplies, and F. O. Glöckner. 2012. SINA: Accurate high-throughput multiple sequence alignment of ribosomal RNA genes. Bioinformatics 28:1823-1829.
19. Redecker, D., A. Schüßler, H. Stockinger, S. Stürmer, J. Morton, and C. Walker. 2013. An evidence-based consensus for the classification of arbuscular mycorrhizal fungi (Glomeromycota). Mycorrhiza:1-17.
20. Ruan, Y., S. Ekanayake, M. Rho, H. Tang, S.-H. Bae, J. Qiu, and G. Fox. 2012. DACIDR: deterministic annealed clustering with interpolative dimension reduction using a large collection of 16S rRNA sequences. Pages 329-336 Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine. ACM, Orlando, Florida.
21. Schloss, P. D., S. L. Westcott, T. Ryabin, J. R. Hall, M. Hartmann, E. B. Hollister, R. A. Lesniewski, B. B. Oakley, D. H. Parks, C. J. Robinson, J. W. Sahl, B. Stres, G. G. Thallinger, D. J. Van Horn, and C. F. Weber. 2009. Introducing mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities. Applied and Environmental Microbiology 75:7537-7541.
22. Schoch, C. L., K. A. Seifert, S. Huhndorf, V. Robert, J. L. Spouge, C. A. Levesque, W. Chen, and C. Fungal Barcoding. 2012. Nuclear ribosomal internal transcribed spacer (ITS) region as a universal DNA barcode marker for Fungi. Proceedings of the national academy of sciences 109:6241-6246.
23. Stamatakis, A. 2006. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics 22:2688-2690.
24. Sun, Y., Y. Cai, L. Liu, F. Yu, M. L. Farrell, W. McKendree, and W. Farmerie. 2009. ESPRIT: estimating species richness using large collections of 16S rRNA pyrosequences. Nucleic Acids Research 37:e76.
25. Ye, Y. 2010. Identification and quantification of abundant species from pyrosequences of 16S rRNA by consensus alignment. Pages 153-157 in Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on. IEEE.
26. Bae, S.-H., J. Y. C., et al. (2010). "Dimension reduction and visualization of large high-dimensional data via interpolation." Proceedings of the 19th ACM HPDC Conference, Chicago, Illinois, ACM.