Chapter 3

Development of Methodology for the Construction of Multiparametric Phylogenetic Trees of Vertebrate Lens Crystallins in 3D.

Introduction

In the chapter 2, I have constructed three uniparametric phylogenetic trees for eight vertebrates each comparing the probability of number of changes per protein based on their distribution in MW, pI and immunocrossreactivity and demonstrated that uniparametric trees for different parameters but the same set of species differ in their topology significantly. Therefore, I have developed a novel approach to construct a consensus phylogenetic tree by combining these 3 parameters for crystallins from those eight vertebrates.

Methodology

Data

The similarity coefficients and the P-values for the three parameters MW, pI and IC used here have been shown (Tables 2.1) and were used to construct multiparametric consensus phylogenetic trees.
Construction of 3D Plots

In order to obtain consensus distances for three parameters between each pair of species, I have used 3-D plots with p values (0.0 to 1.0) for number of changes in protein based on distribution of MW (X axis), pI (Y axis) and IC (Z axis). I then use one among 8 species as reference species at the position (0, 0, 0) and then the remaining in the 3-D plot against their coordinates and repeat the operation with each species as the reference. Thus, for shark as the reference specie, I plot the p values between shark to shark for IC, pI and MW at (0, 0, 0), respectively. Then the p values between the shark and the pomfret (see, Table 2.1) are 0.597 (IC), 0.414 (pI) and 0.536 (MW) and plotted accordingly in the same 3D plot. (Figure 3.1 a). Thereafter, all remaining species paired against the shark are plotted against the reference shark, with the Frog at 0.501, 0.740, 0.693, *Calotes* at 0.501, 1.277, 0.462, the Chick (0.282, 0.903, 0.418), Mouse (0.536, 0.693, 0.536), *Rousettus* (0.327, 0.732, 0.649) and *Megaderma* at 0.501, 0.672, 0.536. This gives a 3D plot for 8 vertebrates with Shark as the reference specie at the origin (see Figure 3.2 a). I then repeat the process with pomfret as the reference species (Figure 3.2 b) to obtain the position of remaining 7 vertebrates in the 3-D plot. (Figure 3.2 b). This process is then repeated for the remaining six species with each as the reference to generate six other 3D plots. (Figure 3.1 a-h).
Figure 3.1: 3D plots for 8 vertebrates. a) Shark b) Pomfret c) Frog d) Calotes e) Chick f) Mouse g) Rousettus h) Megaderma. In each plot one specie is positioned at the origin as reference and the remaining species are positioned in the 3D space.
Euclidian Distance Estimation

The linear distance between any two objects in a 3D space is the Euclidean distance that is estimated using the formula (Wikipedia, Euclidean distance — Wikipedia, The Free Encyclopedia, 2009):

\[ D = \sqrt{(X_1 - X_2)^2 - (Y_1 - Y_2)^2 - (Z_1 - Z_2)^2} \]

Where, \( D \) is the Euclidian distance between the objects located at \( (X_1, X_2) \) and \( (X_1, X_2) \).

From the eight X, Y, Z plots (Figure 3.1), I estimated the set of Euclidian distances between each pair of species with that specie at the origin and the remaining seven in the 3D space in each of 8 plots. These distances are named as 'star' Euclidian distances (see Table 1.1). I then estimated the Euclidian distances among the seven species pairs (excluding the reference specie) in the 3D space of each plot and these are termed as the 'network' Euclidian distances. The mean Euclidean 'star + network' distances (see Table 1.3), along with standard deviation.
### Table 3.1 'Star' distance among individual species

<table>
<thead>
<tr>
<th>Distance</th>
<th>Shark</th>
<th>Pomfret</th>
<th>Frog</th>
<th>Calotes</th>
<th>Chick</th>
<th>Mouse</th>
<th>Rousettus</th>
<th>Megaderma</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shark</td>
<td>0</td>
<td>0.82 ± 0.11</td>
<td>1.08 ± 0.07</td>
<td>1.50 ± 0.08</td>
<td>1.22 ± 0.27</td>
<td>0.97 ± 0.09</td>
<td>1.14 ± 0.15</td>
<td>1.00 ± 0.00</td>
</tr>
<tr>
<td>Pomfret</td>
<td>0.82 ± 0.11</td>
<td>0</td>
<td>0.85 ± 0.00</td>
<td>0.96 ± 0.06</td>
<td>1.19 ± 0.19</td>
<td>0.85 ± 0.09</td>
<td>1.00 ± 0.06</td>
<td>1.07 ± 0.13</td>
</tr>
<tr>
<td>Frog</td>
<td>1.08 ± 0.07</td>
<td>0.85 ± 0.00</td>
<td>0</td>
<td>0.89 ± 0.03</td>
<td>0.70 ± 0.00</td>
<td>0.76 ± 0.00</td>
<td>0.93 ± 0.06</td>
<td>0.82 ± 0.12</td>
</tr>
<tr>
<td>Calotes</td>
<td>1.50 ± 0.08</td>
<td>0.96 ± 0.06</td>
<td>0.89 ± 0.03</td>
<td>0</td>
<td>0.68 ± 0.00</td>
<td>0.98 ± 0.06</td>
<td>0.91 ± 0.06</td>
<td>0.91 ± 0.11</td>
</tr>
<tr>
<td>Chick</td>
<td>1.22 ± 0.27</td>
<td>1.19 ± 0.19</td>
<td>0.70 ± 0.00</td>
<td>0.68 ± 0.00</td>
<td>0</td>
<td>1.08 ± 0.17</td>
<td>1.07 ± 0.00</td>
<td>0.94 ± 0.09</td>
</tr>
<tr>
<td>Mouse</td>
<td>0.97 ± 0.09</td>
<td>0.85 ± 0.09</td>
<td>0.76 ± 0.00</td>
<td>0.98 ± 0.06</td>
<td>1.08 ± 0.17</td>
<td>0</td>
<td>0.14 ± 0.05</td>
<td>0.16 ± 0.00</td>
</tr>
<tr>
<td>Rousettus</td>
<td>1.14 ± 0.15</td>
<td>1.00 ± 0.06</td>
<td>0.93 ± 0.06</td>
<td>0.91 ± 0.06</td>
<td>1.07 ± 0.00</td>
<td>0.14 ± 0.05</td>
<td>0</td>
<td>0.16 ± 0.12</td>
</tr>
<tr>
<td>Megaderma</td>
<td>1.00 ± 0.00</td>
<td>1.07 ± 0.13</td>
<td>0.82 ± 0.12</td>
<td>0.91 ± 0.11</td>
<td>0.94 ± 0.09</td>
<td>0.16 ± 0.00</td>
<td>0.16 ± 0.12</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 3.2 'Network + star' distance among individual species

<table>
<thead>
<tr>
<th>Distance</th>
<th>Shark</th>
<th>Pomfret</th>
<th>Frog</th>
<th>Calotes</th>
<th>Chick</th>
<th>Mouse</th>
<th>Rousettus</th>
<th>Megaderma</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shark</td>
<td>0.67 ± 0.31</td>
<td>0.45 ± 0.38</td>
<td>0.52 ± 0.30</td>
<td>0.60 ± 0.31</td>
<td>0.58 ± 0.35</td>
<td>0.11 ± 0.14</td>
<td>0.13 ± 0.08</td>
<td>0</td>
</tr>
<tr>
<td>Pomfret</td>
<td>0.45 ± 0.38</td>
<td>0.52 ± 0.30</td>
<td>0.60 ± 0.31</td>
<td>0.58 ± 0.35</td>
<td>0.11 ± 0.14</td>
<td>0.13 ± 0.08</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Frog</td>
<td>0.52 ± 0.30</td>
<td>0.60 ± 0.31</td>
<td>0.58 ± 0.35</td>
<td>0.11 ± 0.14</td>
<td>0.13 ± 0.08</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Calotes</td>
<td>0.60 ± 0.31</td>
<td>0.58 ± 0.35</td>
<td>0.11 ± 0.14</td>
<td>0.13 ± 0.08</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chick</td>
<td>0.58 ± 0.35</td>
<td>0.11 ± 0.14</td>
<td>0.13 ± 0.08</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mouse</td>
<td>0.11 ± 0.14</td>
<td>0.13 ± 0.08</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rousettus</td>
<td>0.13 ± 0.08</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Megaderma</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Phylogenetic Tree Construction

To construct consensus phylogenetic trees, I have used distances matrices for 'Star' and 'network + star' Euclidian distances (see Tables 3.1 and 3.2). The trees were constructed by UPGMA method as described earlier (see, chapter 2) using PHYLIP (Felsenstein, 1989) software package.

Bootstrapping phylogenetic trees

Bootstrap analysis is an advanced statistical method used to estimate the confidence in the raw data. I have performed bootstrap analysis to obtain the Confidence Limits (CL) for each tree. Typically, 1000 distance matrices were constructed using Box-Muller (Box & Muller, 1958) method. In this method, initially two random numbers \((U_1\) and \(U_2\)) were assigned a value between 0 and 1, while the Variables \(X\) and \(Y\) were calculated using the formulae:

\[
X_1 = (-2\log U_1)\sin(2\pi U_2) \quad \& \quad X_1 = (-2\log U_1)\sin(2\pi U_2)
\]

The bootstrap sample distances were calculated by the equation:

\[
X_1 = sX + m \quad \& \quad Y_1 = sY + m,
\]

Where, \(s\) is the standard deviation, \(m\) is the mean and \(X_1\) & \(Y_1\) are the bootstrap distances. Thus, 1000 distances (500 pairs) were generated and subjected to UPGMA. The topologies of bootstrap sample were compared to the topology of the tree obtained from mean 'star' Euclidian distances or 'network' Euclidian distances and the confidence limits for each tree are calculated as:

\[
CL = P + 1.96\sigma,
\]

Where, CL is the confidence limit, \(P\) is the probability of observing trees similar to mean distances tree and \(\sigma\) is calculated using the formula
\[ \sigma = \left( \frac{(P(1-P)/n)}{\sigma} \right)^{1/2} \]

for ‘n’ samples.

**Multidimensional Scaling**

Multidimensional scaling (MDS) (Cox, 2000) is a collection of techniques for data analysis to estimate coordinates from the distance matrix in desired number of dimensions, typically in a 3-D space. In MDS, the objects are efficiently rearranged to get a configuration that best approximates the observed distances. If there are ‘n’ number of objects and the distances between \( i^{th} \) and \( j^{th} \) (i, j=1 to n) objects are known, then MDS can fix these objects in desired dimensions (at least 3). It actually moves objects around in the space defined by the required number of dimensions, and checks how well the distances between objects can be reproduced by the new configuration. The algorithm is as follows:

**INPUT:** Input to MDS is matrix of distances / dissimilarities between \( i^{th} \) and \( j^{th} \) objects.

**ALGORITHM**

Assign points to arbitrary coordinates in the lower dimensional space.

Compute Euclidean distances among all pairs of points, to form the D’ matrix.

Compare the D’ matrix with the input D matrix by evaluating the stress function. The most common measure that is used to evaluate a particular configuration reproduces the observed distance matrix is the stress measure. The smaller the value, the greater is the correspondence between the matrices D’ and D.

Adjust coordinates of each point in the direction that minimizes stress function.

Repeat steps 2 through 4 until the stress is not reduced any further.
Construction of 3D Phylogenetic Trees

Two dimensional tree building is a very well established technique (Kidd & Sgaramella-Zonta, 1971; Rzhetsky & Nei, 1993; Saitou & Nei, 1987; Sneath & Sokal, 1962; Wikipedia, Computational phylogenetics — Wikipedia, The Free Encyclopedia, 2009; Wikipedia, Heuristic algorithm — Wikipedia, The Free Encyclopedia, 2009; Wikipedia, Maximum parsimony — Wikipedia, The Free Encyclopedia, 2009) and several software tools (Felsenstein, 1989; Galtier, Gouy, & Gautier, 1996; Maddison & Maddison, 2006; Rice, Longden, Bleasby, & others, 2000; Swofford, 2002; Tamura, Dudley, Nei, & Kumar, 2007; Xia & Xie, 2001) are available for this purpose. I have used the most popular tool PHYLIP (Felsenstein, 1989) to construct 2D phylogenetic trees. However, there is no methodology available to construct multiparametric phylogenetic trees visualized in 3D, and I have addressed this problem in this thesis.

If I process the Euclidian distance matrix (see Table 1.1, 1.3) with MDS for 3D phylogenetic tree building, then I obtain the position of the terminal nodes (species) but not the internal nodes (ancestral nodes) position. In order to obtain all pairs distances for both terminal and internal nodes, I have developed an all pairs shortest distance algorithm (APSDA). In 2D phylogenetic tree (see Figure 3.2), the only known distance is between any node and their immediate ancestor. As shown in the Figure 3.1, A1 is the ancestor for T1 & T2 and A2 is the ancestor for T3 & T4. The node R is the root of the tree which is also the ancestor for the internal nodes A1 & A2. Here, the distances between the pairs (T1, A1), (T2, A1), (T3, A2), (T4, A2), (A1, R), (A2, R) are known, while the distances between the nodes across the dendrogram, (e.g., pairs [A1, A2], [T1, T4]), are not known. In practice, one can traverse the tree and estimate the distance between nodes manually, but with increasing
number of nodes, this process becomes unwieldy. Therefore, I have developed the algorithm APSDA to estimate the unknown distances among all nodes (e.g., both terminal & internal).

![Phylogenetic Tree Diagram](image)

**Figure 3.2**: A schematic diagram of a phylogenetic tree. The nodes T1, T2, T3 & T4 are the terminal nodes or species. A1 is the ancestral node for T1 & T2 and A2 is the ancestor for T3 & T4. The node R is the root of the tree as well as ancestral node for A1 & A2.

**All Pairs Shortest Distance Algorithm (APSDA)**: With “n” as the total number of terminal nodes/species and an unrooted phylogenetic tree contains “n-2” internal nodes, while rooted trees has “n-1” internal nodes. Therefore, I get the total number of nodes as “2n-2” for unrooted and “2n-1” for rooted trees. If I use a rooted phylogenetic tree with 8 terminal nodes, I need a distance matrix with [(2x8)-2] x [(2x8)-2], or 15x15 for all pairs. I have developed All Pairs Shortest distance Algorithm (APSDA) that can handle an incomplete distance matrix as input in which the distance between any node and its immediate ancestor, but not a distant node, is known. APSDA iteratively estimates the unknown values till the
distance matrix is complete. Thus, treatment with APSDA, the distances from any pairs of nodes are found. The detail description of this algorithm is given below.

Using APSDA, I have constructed a distance matrix (D) of size 'N x N' for 'N' number of nodes which I have filled with known distances between any nodes. Then I set the unknown values to NULL in the distance matrix. This algorithm iterates through all rows and finds all possible known distance pairs. New distances are computed from these distance pairs and entered in the distance matrix 'D'. The algorithm terminates when the 'D' matrix is completely filled.

Pseudocode

**Input:** List of edges/branches and costs/distances in a phylogenetic tree.

**Output:** Distance matrix (D).

**Initialization:** Construct a distance matrix D of size 'N x N' where 'N' is the number of nodes. Initialize D[i,j] to known distance between adjacent nodes 'i' and 'j'. Set D[i,j] to 'NULL' if not 'i' and 'j' are not adjacent nodes.

First define a function GET_UNIQUE_COMBINATIONS, which takes a row of a matrix as argument. The output will be all possible unique pairs (i, j) such that array[i] and array[j] are not NULL.

Function GET_UNIQUE_COMBINATIONS(array)

Return all possible unique pairs (i, j)

where array[i] and array[j] are not NULL

End Function GET_UNIQUE_COMBINATIONS

Next, define the function GENERATE_DISTANCE_MATRIX that takes the initialized distance matrix D as its argument. The function processes one row at a time using
the index 'row_index' of the row. For each row, unique pairs of node indices are generated using the function GET_UNIQUE_COMBINATIONS. If no combinations are returned by the function then the algorithm terminates. Otherwise, for each index combination \((i_1, i_2)\) the new distance is computed by the equation \(D[i_1,i_2]=D[row_index][i_1]+ D[row_index][i_2]\).

Function GENERATE_DISTANCE_MATRIX(D)

For row_index = 1 to N

Result=GET_UNIQUE_COMBINATIONS(D[row_index])

If (length of result is 0)

Exit

Else

For every pair \((n_1, n_2)\) in result

\(D[n_1,n_2]=D[row_index][n_1]+ D[row_index][n_2]\)

Return D

End function GENERATE_DISTANCE_MATRIX

As I have estimated all pairs shortest distance between terminal as well as internal nodes using APSDA, I am able to use it as input to place all nodes in a 3-D space in order to construct a phylogenetic tree in 3-D. The task of positioning nodes in the 3-D space was accomplished by applying the method of multi-dimensional scaling (MDS).

**Results and Discussion**

First, using Euclidean distance matrix as the input, I have constructed multiparametric phylogenetic trees in 2D for eight vertebrates (Figure 3.3 a-b). Unlike uniparametric trees for these 8 species (Figure 2.1-3), the 2D multiparametric tree shows that Mammals, fishes and other 3 vertebrates (frog, lizard and chick) always form a discrete cluster, but their relative
positions were different for “star” (Figure 3.3a, 3.4a) and “network + star” (Figure 3.3b, 3.4b) Euclidian distances. I conclude that the multiparametric phylogenetic trees give a representation that is taxonomically meaningful. I have also performed Bootstrap analysis to obtain the Confidence Limits (CL) which were 0.48 + 0.09 for “star” Euclidean distance, and 0.82 + 0.07 for “network + star” Euclidian distances. I tentatively conclude that “network + star” Euclidian distances give better fit, which, however, may not hold for a large number of species because the network distances give varying distance values while “star” distances do not.

Figure 3.3. Multiparametric phylogenetic trees obtained by estimating Euclidian distances. a) 'star'- b) ‘network + star'- Euclidian distance
In the existing 2D phylogenetic tree the output distance matrix for all pairs distances is incomplete. This issue becomes even further exacerbated for multiparametric trees. Among the principle methods available at present to compare the position of nodes, the simplest one is Tree Traversal Algorithm (TTA) (Dale & Lilly, 1995; Drozdek, 2004; Wikipedia, Tree traversal — Wikipedia, The Free Encyclopedia, 2009) which assumes each node to be either binary 1.2 or equally polytomous and does not cater to the situation involving random binary and polytomous dendrogram. This algorithm is not applicable to multidirectional/circular network. Another important method Floyd–Warshall Algorithm (FWA) (Cormen, Leiserson, Rivest, & Stein, 2001; Floyd, 1962; Kleene, 1951; Rosen, 1999; Warshall, 1962; Wikipedia, Floyd–Warshall algorithm — Wikipedia, The Free Encyclopedia, 2009) which caters for complex networks and dendrograms including randomly distributed binary and polytomous
nodes. However, this algorithm does not accept negative values that are often generated using neighbor-joining method. Furthermore, this algorithm is computationally very expensive. In a standard phylogenetic tree, one may obtain both binary and polytomous nodes that are randomly distributed along the three axes and that may have negative branch lengths in case of NJ trees. Indeed in most of my work in the following chapters uses NJ trees. Therefore, I have developed a new algorithm to estimate all pairs shortest distances. Indeed, this algorithm allows completion of distance matrix for both uniparametric and multiparametric tree topologies in 2D as well as 3D.
Figure 3.3: Representation of phylogenetic trees (a) A binary phylogenetic tree with 'a', 'b', 'c' & 'd' as terminal nodes, 'n1' & 'n2' as internal nodes and 'r' as the root. (b) A polytomyous phylogenetic tree in which the node 'n1' having two children but the node 'n2' having three children. (c) A network type phylogenetic tree. Here the node 'n2' is connected to 'n1', 'a', 'c', 'd' & 'e'. Similarly 'n1' is connected to 'a', 'b', 'd' & 'n1'. This relationship clearly shows a complex network among species and their ancestors.

I have also examined the 3D positioning of the eight species on the basis of Mahalanobis distance (Wikipedia, Mahalanobis distance — Wikipedia, The Free Encyclopedia, 2009), which is measured in an ellipsoidal space. The phylogenetic trees (not shown) based on Mahalanobis distances show that the error bars were between 30 and 50%. Thus, the present data do not fall under Mahalanobis coordinates. However, for a heterogeneous population of organisms, this method may become useful.

In this chapter I have used only eight representatives of the subphylum vertebrata that allow distinction among classes, subclasses in general and up to suborder in mammals. Clearly, a phylogenetic tree based on multiple parameters is closer to taxonomic reality than that based on a single trait/character/property.
In conclusion, high-resolution 3D phylogenetic trees can be constructed using biochemical (pI and MW), immunochemical (IC) parameters or any other quantifiable character/trait to give discrete phylogenetic relationships based on multiple parameters/traits.