계통발생학적 분석을 위한 분류 단위의 제약조건 기반의 3차원 시각화 기법

A Constraint-based Three-Dimensional Visualization Method of Operational Taxonomic Units for Phylogenetic Analysis

이선아, 이건명

Sun-a Lee, Keon Myoung Lee
School of Electrical and Computer Engineering, Chungbuk National University
E-mail : salee719@aicore.cbnu.ac.kr

요 약

계통발생학적 분석기법은 서열의 유사성을 비교하여 이들의 유연관계를 알아내는 것으로, 각각의 관계를 시각적으로 표현하는 것이 매우 중요하다. 일반적으로 2차원 계통수를 사용한다. 그러나 2차원으로 시각화했을 때 서로 유사성이 높은 OTU(Operational Taxonomic Unit)들을 서로 멀리 떨어뜨려 놓는 경우도 생기게 된다. 이 논문에서는 이러한 점을 보완하고자 3차원 공간에 OTU들을 배치시키기 위한 2단계 좌표 배치 기법을 제안한다. 1단계는 유클리디안 거리를 3차원 좌표로 변환하는 것이다. 1단계 방법은 서열의 비교 순서에 영향을 받기 때문에 2단계를 통해 유전자 알고리즘 기법을 적용하여 보다 적절한 좌표를 찾는다.

1. 서론

계통발생학적 분석기법은 서열의 유사성을 비교하여 이들의 유연관계를 알아내는 것으로, 이들 관계를 시각적으로 나타내는 것이 매우 중요하다. 계통수 연구를 통해 바이러스의 근원, 변천과정 등을 유추할 수 있다. 일반적으로 2차원 계통수를 사용한다. 계통수를 얻는 방법은 문자 위주의 방법(character-based method)과 거리 기반의 방법(distance-based method)이 있다. 문자 위주의 방법은 MP(Maximum Parsimony) 방법과 ML(Maximum Likelihood) 방법이 있다. 거리 기반의 방법은 UPGMA(Unweighted Pair Group Method with Arithmetic mean), NJ(Neig-hber joining), FM(Fitch-Margolish) 방법 등이 있다. 거리 기반의 방법이 많이 사용이 되는데 이 방법은 먼저 모든 서열들 사이의 거리를 계산한다. UPGMA 방법은 OTU들 사이에서 두 개의 가장 근접한 OTU를 선택한다. 가장 근접한 것은 가장 유사한 것을 의미하며 공통 조상을 가진다. 하지만 이러한 방법들은 사용할 때의 배열과 같은 환경적인 요소들로 인해서 적용할 때의 환경이 영향을 끼친다. 또한 2차원으로 표현하기 때문에 모든 정보를 원하는 만큼 표현하는 것은 쉽지 않다. 2차원 상으로는 서로 떨어져 있지만 다른 정보를 더 표현하게 되면 실제로는 가까운 거리에 있을 수 있다. 때문에 이 논문에서는 3차원으로 시각화하여 표현하여 보다 더 많은 정보를 표현할 수 있도록 하고자 한다.

이 논문에서는 2장에서 더 많은 정보를 표현하기 위한 시각화 방법에 관련된 연구를 소개하고 3장에서는 제안한 3차원 시각화 방법을 설명하고 4장에서 구현된 내용을 간단히 보인다.

2. 관련 연구

계통수를 시각화하는 방법은 시각화 접근 방법

이 논문은

[1], 시각화를 돕는 여러 가지 기본 도구들, 그리고 3차원으로 접근한 방법 등 그 연구 분야가 다양하다. 시각화 접근 방법[1]은 크게 5가지로 소개하고 있다. 레이아웃(Layout), 라벨 및 주석 붙이기(Labeling and Annotation), 트리보기(Navigation), 트리비교(Tree Comparision), 조작 및 수정(Manipulation and editing)을 포함한다. 레이아웃은 계통수를 표현하는 네트워크 구조를 연구하는 것이다. 라벨 및 주석 붙이기는 계통수를 표현한 다음 각각의 내용을 한눈에 보기 쉽도록 계통수 내에 각각의 이름을 표현한다. 하지만 이것은 복잡하고 많은 내용을 표현하고자 할 경우에는 오히려 내용이 겹쳐서 보이는 등의 단점이 있으므로 고려할 사항이 많다. 트리보기의 경우에는 많은 내용을 사용자가 보기 쉽도록 조작할 수 있도록 하는 것을 연구하는 것이다. 트리보기에는 전체를 한 눈에 보는 방법(Overview), 특정 내용을 보는 방법은 줌과 필터링(Zoom and Filter), 마지막으로 사용자가 원하는 정보만 보는 방법(detail-on-demand)가 있다. 트리비교의 경우에는 사용자가 몇 가지 트리를 한눈에 보고 비교하는 기능을 제공한다. 조작 및 수정은 사용자가 출판하기 위한 보고서를 만드는 것을 돕는다.

계통수를 시각화하는 것을 돕는 어플리케이션으로는 가장 쉽게 사용할 수 있는 PAUP나 PHYLIP과 같은 것이 있다. TreeWiz[2]와 같은 어플리케이션은 여러 개의 윈도우를 사용하여 트리를 탐색하도록 한다. TreeJuxtapose[3]와 같은 어플리케이션은 많은 수의 노드의 구조를 서로 비교할 수 있도록 한다.

2차원 시각화 도구 외에도 3차원 시각화 도구를 개발하는 예가 있다. 예로 Arbor 3D[4]와 같은 어플리케이션이 있다. Arbor 3D는 3차원으로 계통수를 시각화하여 그 내용을 가상공간 상에서 보여주어 사용자가 가상공간 장비를 사용하여 그 내용을 편집할 수 있도록 한다.

이 논문에서는 계통수를 3차원으로 시각화하기 위하여 좌표값을 구하여 그 값을 실제 화면에 보여주는 방법을 제안한다.

3. 제안한 시각화 방법

이 장에서는 계통수의 데이터인 OTU를 3차원으로 시각화하기 위하여 3차원 좌표를 구하는 방법을 제안한다. 먼저 몇 가지 내용을 정의한다.

$S_i : i-th\ OTUs(\ i\ =\ 1,\ ...,\ n)$

n : 좌표값을 구한 OTU의 개수

$P_i = (x_i,\ y_i,\ z_i)$: S_i좌표와 일치하는 3차원 상의 좌표

$\delta(S_i, S_j)$: 서열 정렬(*sequence assignment*)와 같은 평가 방법을 이용해 얻은 S_i와 S_j사이의 거리

$d(P_i,\ P_j)$: 3차원 상의 P_i *and* P_j 의 거리

$\alpha_i = d^2(P_i, P_j)$

3차원으로 표현할 때의 문제점은 서열 정렬과 같은 방법으로 구한 유클리디안 값을 3차원의 좌표값 $P_i = (x_i, y_i, z_i)$ 으로 변환하는 방법이 필요하다. 이 때에 P_i값을 구하기 위해 다음과 같은 제안을 둔다.

$$d(P_i, P_j) = \delta(S_i, S_j) \qquad (i \le i, j \le n)$$

3.1 유클리디안 거리에 대한 좌표 배치 방법

유클리디안 공간상의 값을 속성이 그대로 유지하면서 3차원 좌표값으로 바꾸기 위해 우리는 다음과 같은 제한을 두었다. 먼저 P_1은 3차원 공간의 원점에 배치하고 P_2는 y와 z축을 고정하고 x축의 좌표를 구한다. 즉 x좌표가 $d(P_2, P_1)$ 이다. P_3은 z축을 고정한 채, xy-평면에 그 값을 구한다. 이 경우 $d(P_3, P_1) = \delta(S_3, S_1)$ 과 $d(P_3, P_2) = \delta(S_3, S_2)$ 를 모두 만족해야 한다. n개의 OTU의 좌표를 구하기 위해서는 P_4 좌표값부터는 축을 고정시키지 않고 구한다. 때문에 4번째 좌표부터는 좌표값이 충돌을 일으키게 된다. 구하는 것을 수식으로 표현하면 다음과 같다.

$P_1 = (x_1, y_1, z_1) = (0,0,0)$

$P_2 = (x_2, y_2, z_2) = (d(P_2, P_1), 0, 0)$

$P_3 = (x_3, y_3, z_3)$

$y_3 \ge 0, z_3 = 0$

$d^2(P_3, P_1) = x_3^2 + y_3^2 + z_3^2 = \alpha_3$

$$\begin{aligned} d^2(P_3, P_2) &= (x_3 - x_2)^2 + (y_3 - y_2^2) + (z_3 - z_2)^2 \\ &= x_3^2 + y_3^2 + z_3^2 + x_3^2 + y_3^2 + z_3^2 - 2x_2x_3 - 2y_2y_3 - 2z_2z_3 \\ &= \alpha_3 + \alpha_2 - 2x_2x_3 - 2y_2y_3 - 2z_2z_3 = \delta(S_3, S_2) \end{aligned}$$

$P_4 = (x_4, y_4, z_4)$

$d^2(P_4, P_1) = x_4^2 + y_4^2 + z_4^2 = \alpha_4$

$d^2(P_4, P_2) = \alpha_4 + \alpha_2 - 2x_2x_4 - 2y_2y_4 - 2z_2z_4 = \delta(S_4, S_2)$

$d^2(P_4, P_3) = \alpha_{i4} + \alpha_3 - 2x_3x_4 - 2y_3y_4 - 2z_3z_4 = \delta(S_4, S_3)$

n개를 구하기 위해 식을 일반화하면 다음과 같다.

$P_i = (x_i, y_i, z_i)$

$d^2(P_i, P_1) = x_i^2 + y_4^2 + z_4^2 = \alpha_4$

$d^2(P_i, P_2) = \alpha_i + \alpha_2 - 2x_2x_i - 2y_2y_i - 2z_2z_i$
$= \delta(S_i, S_2)$

$d^2(P_i, P_3) = \alpha_i + \alpha_3 - 2x_3x_i - 2y_3y_i - 2z_3z_i$
$= \delta(S_i, S_3)$

• • •

3.2 좌표 배치 문제의 완화

모든 제약조건을 만족하는 OTU의 3차원 공간 좌표를 구하는 것은 쉽지 않다. 이유는 조건이 많아지면 많아질수록 충돌이 일어나는 부분이 많이진다. 이 논문에서는 이 부분을 조정하기 위해 제약조건을 만족하는 좌표값을 구하고자 하는 방법을 제안하고자 한다. 모든 조건을 만족하는 것은 불가능하기 때문에 이를 완화하기 위해 우리는 Lagrangian 최적화 방법을 적용하였다. 먼저 풀고자 하는 문제를 수식화하면 다음과 같다.

Find $P_i = (x_i, y_i, z_i)$
which $\min imizes (x_i^2 + y_i^2 + z_i^2 - \alpha_i)^2 +$
$\sum_{j=2}^{i-1} (\alpha_i + \alpha_j - 2x_jx_i - 2y_jy_i - 2z_iz_j - \delta^2(S_i, S_j))^2$
under the assumptions that
$P_{k(x_k}, y_k, z_k), k = 1, \ldots, i-1$ *are given and*
$\alpha_l,\ l = 1, \ldots, i$ *and* $\delta(S_i, S_j), 1 \le i, j \le$
are given

위의 내용을 Lagrangian 함수인 $J_i(x,y,z)$ 로 표현할 수 있다.

$$J_i(x,y,z) = f_i(x,y,z) + \beta_i g_i(x,y,z) + \sum_{j=2}^{i-1} \gamma_j h_{ij}(x,y,z)$$
$$\beta_i \ge 0, \gamma_j \ge 0 (j = 2, \ldots, i-1)$$

where

$f_i(x,y,z) = (x_i^2 + y_i^2 + z_i^2 - \alpha_i)^2 +$
$\sum_{j=2}^{i-1} (\alpha_i + \alpha_j - 2x_jx_i - 2y_jy_i - 2z_jz_i - \delta^2(S_i, S_j))^2$

$g_i(x,y,z) = x_i^2 + y_i^2 + z_i^2 - \alpha_i$

$h_{ij}(x,y,z) = \alpha_i + \alpha_j - 2x_jx_i - 2y_jy_i - 2z_jz_i - \delta^2(S_i, S_j)$

하지만 이 경우 식이 너무 복잡해지기 때문에 이를 완화하기 위해 a_i, b_i, c_i의 제약조건을 추가한다. a_i, b_i, c_i을 $a_i = x_i^2, b_i = y_i^2, c_i = z_i^2$ 로 변환하도록 하였다. 이는 다음과 같이 표현할 수 있다.

$J_i(x,y,z,a,b,c)$
$= f_i(x,y,z,a,b,c) + \beta_i g_i(a,b,c) + \sum_{j=2}^{i-1} \gamma_j h_{ij}(x,y,z) + \eta_{i1} s_{i1}(a,x)$
$+ \eta_{i2} s_{i2}(b,y) + \eta_{i3} s_{i3}(c,z)$
$\beta_i \ge 0, \gamma_j \ge 0 (j = 2, \ldots, i-1), \eta_{ik} \ge 0\ (k = 1,2,3)$

$f(a,b,c) = (a_i + b_i + c_i - \alpha_i)^2 + \sum_{j=2}^{i-1} (\alpha_i + \alpha_j - 2x_jx_i - 2y_jy_i - 2z_jz_i$
$- \delta^2(S_i, S_j))^2$

$g_i(a,b,c) = a_i + b_i + c_i - \alpha_i$

$h_{ij}(x,y,z) = \alpha_i + \alpha_j - 2x_jx_i - 2y_jy_i - 2z_jz_i - \delta^2(S_i, S_j))$

$s_{i1}(a,x) = \eta_{i1}(x_i^2 - a_i)$

$s_{i2}(b,y) = \eta_{i2}(y_i^2 - b_i)$

최적화를 적용하기 위해 MATLAB을 이용하여 값을 구하였다.

3.3 유전자 알고리즘 기반의 순서 결정

3.2에서 수식화하여 구한 값들은 서열 정렬을 구할 때의 환경에 따라 값이 달라질 수 있기 때문에 가장 적절한 값을 구하기 위해 4 단계의 유전자 알고리즘을 적용하였다.

(1) Encoding SCheme 과정 : 순서화된 OTU의 배치 서열을 구한다. 이때에는 서열 정렬 방법을 이용한다.

(2) Genetic Operator 과정 : 순서 기반의 크로스오버와 스왑 과정을 거쳐 여러 가지 해들을 만들어 낸다.

(3) Fitness Evaulation Method 과정 : (2)과정에서 생성된 해들이 적절한 것인지를 판단한다. 이때에는 다음의 식을 이용하여 적절성을 판단한다.

$$F_t(C) = \sum_{1 \le i, j \le n} (d(P_i, P_j) - \delta(S_i, S_j))^2$$

(4) Population Initialization 과정 : 가능한 크기로 가능한한 많이 OTU해를 만들어낸다.

앞의 4 단계를 거쳐 여러 좌표를 원점으로 하여 구한 유클리디안 값을 3차원 좌표로 변환하는 방법을 반복적으로 적용하고 보다 적절한 좌표를 구하도록 한다.

4. 구현

위에서 제안한 방법은 그 계산양이 많기 때문에 대량의 데이터에 적용하고자 할 경우에는 컴퓨터의 성능에 영향을 받는다. 때문에 제안한 논문에서 테스트한 데이터는 데이터의 양에 제한을 두었으며 그래프로 표현할 경우에 라벨을 조작할 수 있도록 하였다. 라벨을 조작하여 사용자가 원

하는 좌표에 해당하는 값만을 볼 수 있도록 하였다.

5. 결론

이 논문은 유클리디안 공간의 좌표값으로 구해진 값을 3차원으로 시각화하기 위해 좌표값을 구하는 방법을 제안하였다. 이때에 2차원에서 3차원으로 공간좌표값이 변환되는 과정에서 일어나는 충돌을 완화하기 위해 Lagrangian 최적화 방법을 적용하였으며 제약조건으로 인해 일어날 수 있는 오류를 적게 하기 위해 유전자 알고리즘을 적용하여 보다 적절한 좌표를 구하고자 하였다. 보다 많은 제약조건을 두기 보다는 더 효율적인 제약조건을 구해야하는 것이 앞으로의 과제이다.

6. 참고문헌

[1] S.F.Carrizo. Phylogenetic Trees : An Information Visualization Perspective. In Proc. the 2nd Asia-Pacific Bioinformatics Conference(APBC2004), Dunedin, New Zealand, 2004.

[2] U.Rost, E.Bornberg-Bauser. TreeWiz : interactive exploration of huge trees. Bioinformatics 18:109-114, 2002.

[3] T.Munzner, F.Guimbretiere, S.Tasiran, L.Zhang, and Y.Zhou. TreeJuxtapose : Scaleable Tree Comparision using Focus+Context with Guaranteed visibility. In Proc of SIGGRAPH2003, 2003.

[4] R. A. Ruths, E.S, Chen, and L.Ellis. Arbor 3D : An interactive environment for examining phylogenetic and taxonomic trees in multiple dimensions. Bioinformatics, 16(11):1003-1009, 2000.