An algorithm for computing the geodesic distance between phylogenetic trees

Anne Kupczok and Steffen Kläre

Center for Integrative Bioinformatics Vienna
Max F. Perutz Laboratories

December 18th, 2007
An algorithm for computing the geodesic distance between phylogenetic trees

Why yet another tree distance?
An algorithm for computing the geodesic distance between phylogenetic trees

Why yet another tree distance?
An algorithm for computing the geodesic distance between phylogenetic trees
Why yet another tree distance?

Robinson-Foulds distance
An algorithm for computing the geodesic distance between phylogenetic trees

Why yet another tree distance?

Branch-score distance
An algorithm for computing the geodesic distance between phylogenetic trees

Why yet another tree distance?
An (unrooted, bifurcating) topology $\mathcal{T}$ for $n$ taxa corresponds to an orthant $\mathbb{R}^{2n-3}_+$. The unit vectors correspond to the $2n - 3$ splits.
An algorithm for computing the geodesic distance between phylogenetic trees

The tree space

- An (unrooted, bifurcating) **topology** \( \mathcal{T} \) for \( n \) taxa corresponds to an **orthant** \( \mathbb{R}^{2n-3} \)
  - The unit vectors correspond to the \( 2n - 3 \) splits
- A **tree** \( T \) with \( n - 3 \) internal and \( n \) external branch lengths is a point in that orthant

![Diagram of a tree in a Euclidean space](image)
An algorithm for computing the geodesic distance between phylogenetic trees

The tree space

- The **tree space** for \( n \) taxa contains all possible topologies
- Its dimension is the number of splits: \( 2^{n-1} - 1 \)
- Topologies are connected by less resolved topologies
- The unique shortest path between two points is called **geodesic**

Now: Geodesic path connecting two weighted trees $T_1$ and $T_2$. 

Different splits:

$S_1 = (AB | CDEF, CD | ABEF, EF | ABCD)$

$S_2 = (AC | BDEF, FD | ABCE, BE | ACDF)$

$d = |S_1| = |S_2| = 3$
Now: Geodesic path connecting two weighted trees $T_1$ and $T_2$
Dimension $d$ is the number of splits only in one tree

Different splits:

$S_1 = (AB|CDEF, CD|ABEF, EF|ABCD)$

$S_2 = (AC|BDEF, FD|ABCE, BE|ACDF)$

$d = |S_1| = |S_2| = 3$
The set of legal topologies

- Legal topologies are $2d$-dimensional binary vectors
- A 1 indicates that a split is present
- All present splits must be compatible
- The topology is maximal (no 1 can be added)

The two given topologies:

$$\mathcal{T}_1 = \begin{pmatrix} 1, \ldots, 1, 0, \ldots, 0 \end{pmatrix}$$
$$\mathcal{T}_2 = \begin{pmatrix} 0, \ldots, 0, 1, \ldots, 1 \end{pmatrix}$$
The set of legal topologies

- Legal topologies are $2d$-dimensional binary vectors
- A 1 indicates that a split is present
- All present splits must be compatible
- The topology is maximal (no 1 can be added)

The two given topologies:

$$\mathcal{T}_1 = \left( \underbrace{1, \ldots, 1}_d, 0, \ldots, 0 \right), \quad \mathcal{T}_2 = \left( 0, \ldots, 0, 1, \ldots, 1 \right)$$

Example:

$$S = (AB|CDEF, CD|ABEF, EF|ABCD, AC|BDEF, FD|ABCE, BE|ACDF)$$

Topologies: $(1, 0, 0, 0, 1, 0), (0, 1, 0, 0, 0, 1), (0, 0, 1, 1, 0, 0)$
The set of legal topologies

- Legal topologies are $2d$-dimensional binary vectors
- A 1 indicates that a split is present
- All present splits must be compatible
- The topology is maximal (no 1 can be added)

The two given topologies:

$T_1 = \left( \underbrace{1, \ldots, 1}_{d}, 0, \ldots, 0 \right)$
$T_2 = \left( 0, \ldots, 0, \underbrace{1, \ldots, 1}_{d} \right)$

Example:

$S = (AB|CDEF, CD|ABEF, EF|ABCD, AC|BDEF, FD|ABCE, BE|ACDF)$

Topologies: $(1, 0, 0, 0, 1, 0), \quad (0, 1, 0, 0, 0, 1), \quad (0, 0, 1, 1, 0, 0)$
The directed acyclic graph of legal topologies

Two topologies are connected

\( \iff \)

Some of the first \( d \) splits are removed (\( L \)) and some of the last \( d \) splits are added (\( R \))

\[
(1,1,1,0,0,0) \\
(1,0,0,0,1,0) \\
(0,1,0,0,0,1) \\
(0,0,1,1,0,0) \\
(0,0,0,1,1,1)
\]
An algorithm for computing the geodesic distance between phylogenetic trees
Paths through tree space
Enumeration of legal topologies

The directed acyclic graph of legal topologies

Two topologies are connected

Some of the first $d$ splits are removed ($L$) and
some of the last $d$ splits are added ($R$)

$\begin{align*}
&\begin{array}{c}
(1,1,1,0,0,0) \\
(1,0,0,0,1,0) \\
(0,1,0,0,0,1) \\
(0,0,1,1,0,0) \\
(0,0,0,1,1,1)
\end{array}
\end{align*}$
The directed acyclic graph of legal topologies

Two topologies are connected

Some of the first $d$ splits are removed ($L$) and some of the last $d$ splits are added ($R$)

$L=\{2,3\}$
$R=\{5\}$

$(1,1,1,0,0,0) 
(0,1,0,0,0,1) 
(0,0,1,1,0,0) 
(1,0,0,0,1,0) 
(0,0,0,1,1,1)$

$(0,0,0,0,0,0)$
An algorithm for computing the geodesic distance between phylogenetic trees

Paths through tree space

Enumeration of legal topologies

The directed acyclic graph of legal topologies

Two topologies are connected

Some of the first $d$ splits are removed ($L$) and some of the last $d$ splits are added ($R$)

$L = \{1, \ldots, d\}$

$R = \{d+1, \ldots, 2d\}$

Cone path
The path is **parametrized with constant speed** by a piecewise linear function $g$ with $g(0) = T_1$ and $g(1) = T_2$

For edge $e$ in the DAG: **transition time** $t_e = \frac{\|T_1(L_e)\|}{\|T_1(L_e)\| + \|T_2(R_e)\|}$

(Karen Vogtmann, Technical report, Cornell University)
An algorithm for computing the geodesic distance between phylogenetic trees

Paths through tree space

Computation of the exact path

Transition times

- The path is **parametrized with constant speed** by a piecewise linear function $g$ with $g(0) = T_1$ and $g(1) = T_2$

- For edge $e$ in the DAG: **transition time** $t_e = \frac{\|T_1(L_e)\|}{\|T_1(L_e)\|+\|T_2(R_e)\|}$

(Karen Vogtmann, Technical report, Cornell University)
An algorithm for computing the geodesic distance between phylogenetic trees

Paths through tree space

Computation of the exact path

**Transition times**

- The path is **parametrized with constant speed** by a piecewise linear function $g$ with $g(0) = T_1$ and $g(1) = T_2$

- For edge $e$ in the DAG: **transition time** $t_e = \frac{\| T_1(L_e) \|}{\| T_1(L_e) \| + \| T_2(R_e) \|}$

(Karen Vogtmann, Technical report, Cornell University)

```
11 1
1 1 1
1 1 1
1
```

For a sequence of topologies, the transition times must be increasing $\rightarrow$ some sequences turn out to be **illegal**
The length of the path

- For every legal path in the DAG, the length is computed → geodesic path has shortest length

(1, 1, 1, 0, 0, 0) $\xrightarrow{t=0.24}$ (0, 0, 0, 1, 1, 1)

(1, 1, 1, 0, 0, 0) $\xrightarrow{t=0.2}$ (0, 0, 1, 1, 0, 0) $\xrightarrow{t=0.28}$ (0, 0, 0, 1, 1, 1)
An algorithm for computing the geodesic distance between phylogenetic trees

Paths through tree space

Computation of the exact path

The length of the path

- For every legal path in the DAG, the length is computed → geodesic path has shortest length
- \((1, 1, 1, 0, 0, 0) \xrightarrow{t=0.24} (0, 0, 0, 1, 1, 1)\) \(\|g\| = 1.57\)
- \((1, 1, 1, 0, 0, 0) \xrightarrow{t=0.2} (0, 0, 1, 1, 0, 0) \xrightarrow{t=0.28} (0, 0, 0, 1, 1, 1)\) \(\|g\| = 1.56\)
Computational aspects

- The DAG allows a clever enumeration of topologies

- Transition times are computed when generating an edge

- The number of topologies is exponential in $d$
  $\rightarrow$ The algorithm is worst-case exponential in $d$

- Input trees need not be bifurcating
Approximations

**Linear-time approximations**

- **Lower bound:**
  Branch-score distance: $d = ||T_1 - T_2||$ (no path in tree space)

- **Upper bound:**
  Cone path: edge connecting $T_1$ and $T_2$ directly in DAG
An algorithm for computing the geodesic distance between phylogenetic trees

Approximations

Linear-time approximations

- **Lower bound:**
  Branch-score distance: \( d = \| T_1 - T_2 \| \) (no path in tree space)

- **Upper bound:**
  Cone path: edge connecting \( T_1 \) and \( T_2 \) directly in DAG

The bounds differ at most in a factor of \( \sqrt{2} \)

Comparison of the approximations

- Inparanoid database: orthologs from 20 Metazoa species + yeast outgroup (216 orthologs → ML trees with phyML)
- 118 trees without internal polytomies → 6903 pairs
Comparison of the approximations

- Inparanoid database: orthologs from 20 Metazoa species + yeast outgroup (216 orthologs → ML trees with phyML)
- 118 trees without internal polytomies → 6903 pairs

![Comparison Graph]

- Cone/BS
- Geod/BS
- Cone/Geod

All Splits
Comparison of the approximations

- Inparanoid database: orthologs from 20 Metazoa species + yeast outgroup (216 orthologs → ML trees with phyML)
- 118 trees without internal polytomies → 6903 pairs

**All Splits**

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Cone/BS</th>
<th>Geod/BS</th>
<th>Cone/Geod</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>4</td>
<td>1.2</td>
<td>1.2</td>
<td>1.2</td>
</tr>
<tr>
<td>6</td>
<td>1.3</td>
<td>1.3</td>
<td>1.3</td>
</tr>
<tr>
<td>8</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
</tr>
</tbody>
</table>

**Different Splits**

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Cone/BS</th>
<th>Geod/BS</th>
<th>Cone/Geod</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>8</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>10</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>12</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Summary

- Algorithm for the geodesic path connecting two weighted trees
- Exponential in the number of different splits
- Cone path can be computed in linear time and is a good approximation of the geodesic path
Acknowledgements

- Karen Vogtmann

- CIBIV:
  - Arndt von Haeseler
  - Ingo Ebersberger
  - Gregory Ewing

- WWTF (funding)