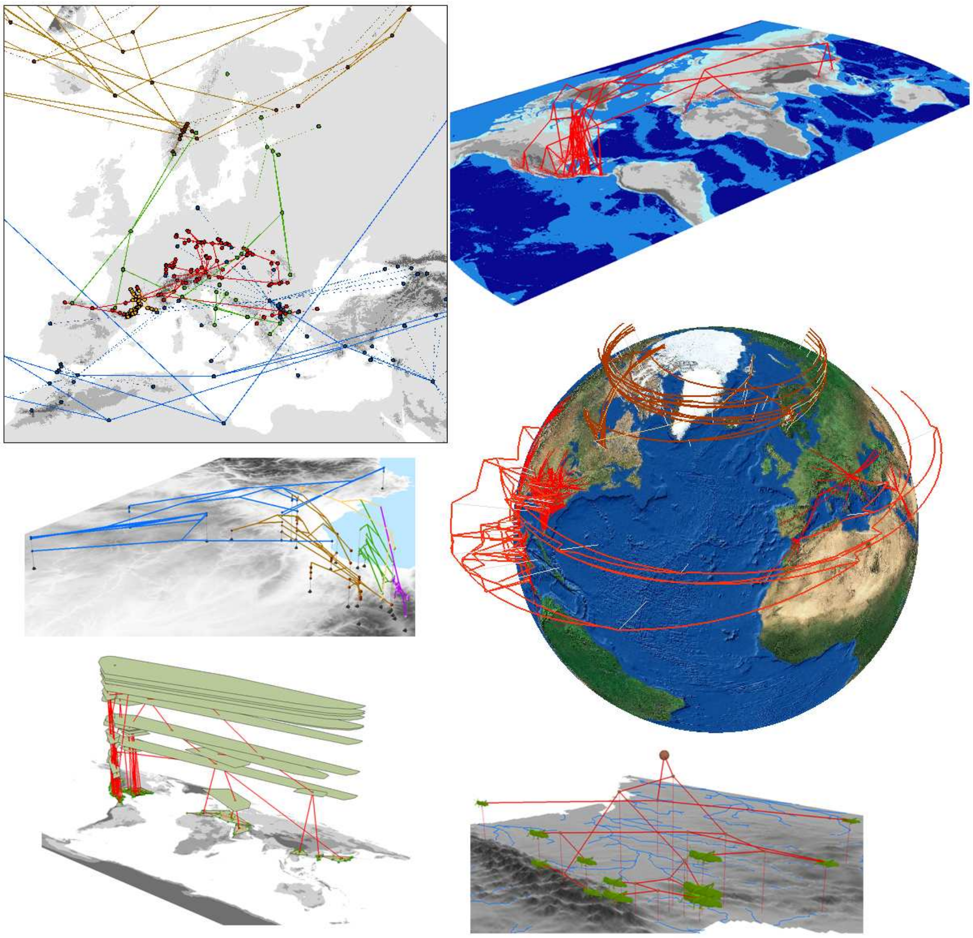
# GeoPhyloBuilder v1.1 for ArcGIS



**David Kidd[1] & Xianhua Liu[2]**

[1]Centre for Population Biology, Imperial College London, Silwood Park Campus

[2]National Evolutionary Synthesis Center, Durham, North Carolina

**February 2010**

**GeoPhyloBuilder**

Copyright (C) 2007, David Kidd.
Copyright (C) 2007, Xianhua Liu, xl24@duke.edu.
Copyright (C) 2007, National Evolutionary Synthesis Center (NESCent).
Copyright (C) 2009-2010, David Kidd, d.kidd@imperial.ac.uk.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

***Supported by the National Evolutionary Synthesis Center, (NESCent)***

***NSF #EF-0423641***

# Contents

# Introduction

Population genetics, phylogeography and historical biogeography combine information on the geographical distribution of biotic variation between individuals, populations, species and higher taxonomic units with information on their relatedness to identify biogeographical pattern and hence infer spatiotemporal evolutionary scenarios. GeoPhyloBuilder is an extension to ESRI's ArcGIS geographical information system (GIS) that builds a 'geophylogeny' (Kidd and Liu, 2008) from an input tree and a set of geographical objects specifying the location of the geographical entities at the tips of the tree. Internal nodes are positioned manually or using spatial averaging algorithms. Tree depth is assigned to the z-attribute of the geophylogeny objects allowing 3D visualization. While developed with phylogenetic data in mind 'geophylogenies' can be created for any entities connected by a tree model.

This Manual describes the GeoPhyloBuilder 1.1 data model, software options and operation. Some example data sets are provided.

# Help, Tutorials and Feedback

Additional information, FAQs, tutorials and example movies are available at https://www.nescent.org/wg_EvoViz/GeoPhyloBuilder.

A set of tutorials can be downloaded from

http://entangled-bank.org/GeophylobuilderTutorial/EvoViz_Workshop.zip

Email geophylobuilder@nescent.org to report bugs and other communications.

# Installation

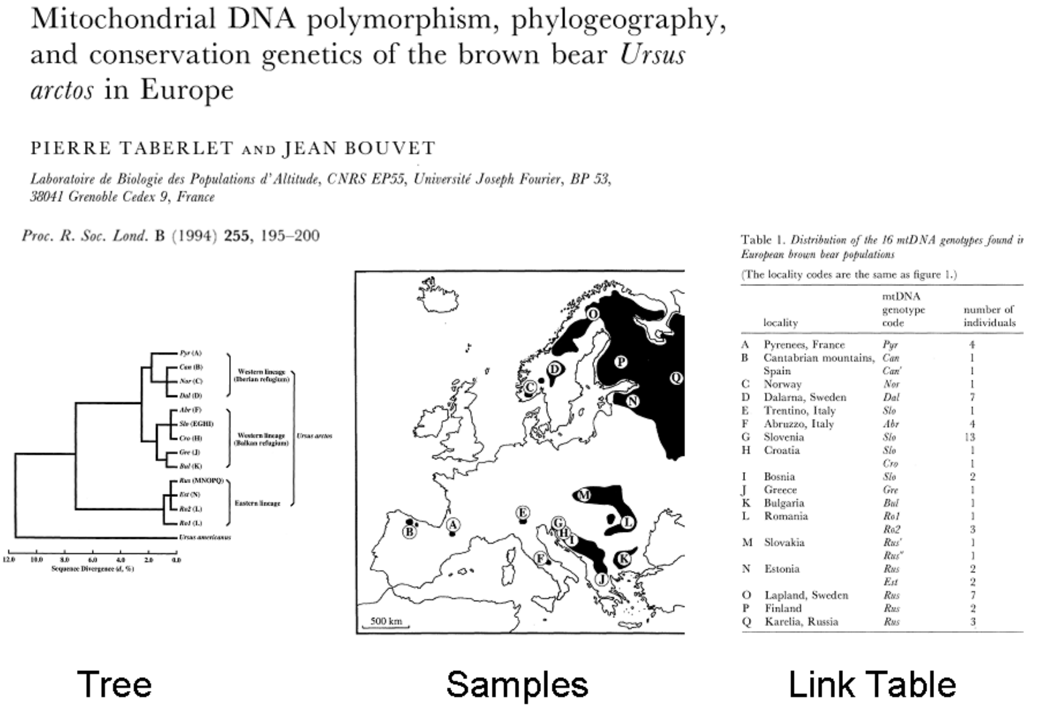
To install double-click *GeoPhyloBuilder.msi* and follow instructions. https://www.nescent.org/wg_EvoViz/GeoPhyloBuilder_Help has information describing various installation issues.

GeoPhyloBuilder is written in .NET using ESRI’s ArcObjects library and a custom library of geophylogenetic COM objects. GeoPhyloBuilder GUI and COM library were developed at the National Evolutionary Synthesis Centre by David Kidd and Xianhua Liu. Source code is freely available on request. We welcome collaboration with users and developers.

## Geophylogenies

Spatial evolutionary data sets are composed of three sets of information (fig 1).

1. *Entities* sampled in space, usually individuals populations or species
2. *A phylogenetic model* (evolutionary tree) defining ‘relatedness’ between entities.
3. A *link table* defining n:m relationships between entities and the tips of the phylogenetic model.



Mitochondrial DNA polymorphism, phylogeography, and conservation genetics of the brown bear *Ursus arctos* in Europe

PIERRE TABERLET AND JEAN BOUVET

*Laboratoire de Biologie des Populations d’Altitude, CNRS EP55, Université Joseph Fourier, BP 53, 38041 Grenoble Cedex 9, France*

*Proc. R. Soc. Lond.* B (1994) **255**, 195–200

Table 1. *Distribution of the 16 mtDNA genotypes found ir European brown bear populations*

(The locality codes are the same as figure 1.)

| | locality | mtDNA genotype code | number of individuals |
|---|---|---|---|
| A | Pyrenees, France | *Pyr* | 4 |
| B | Cantabrian mountains, | *Can* | 1 |
| | Spain | *Can'* | 1 |
| C | Norway | *Nor* | 1 |
| D | Dalarna, Sweden | *Dal* | 7 |
| E | Trentino, Italy | *Slo* | 1 |
| F | Abruzzo, Italy | *Abr* | 4 |
| G | Slovenia | *Slo* | 13 |
| H | Croatia | *Slo* | 1 |
| | | *Cro* | 1 |
| I | Bosnia | *Slo* | 2 |
| J | Greece | *Gre* | 1 |
| K | Bulgaria | *Bul* | 1 |
| L | Romania | *Ro1* | 1 |
| | | *Ro2* | 3 |
| M | Slovakia | *Rus'* | 1 |
| | | *Rus''* | 1 |
| N | Estonia | *Rus* | 2 |
| | | *Est* | 2 |
| O | Lapland, Sweden | *Rus* | 7 |
| P | Finland | *Rus* | 2 |
| Q | Karelia, Russia | *Rus* | 3 |

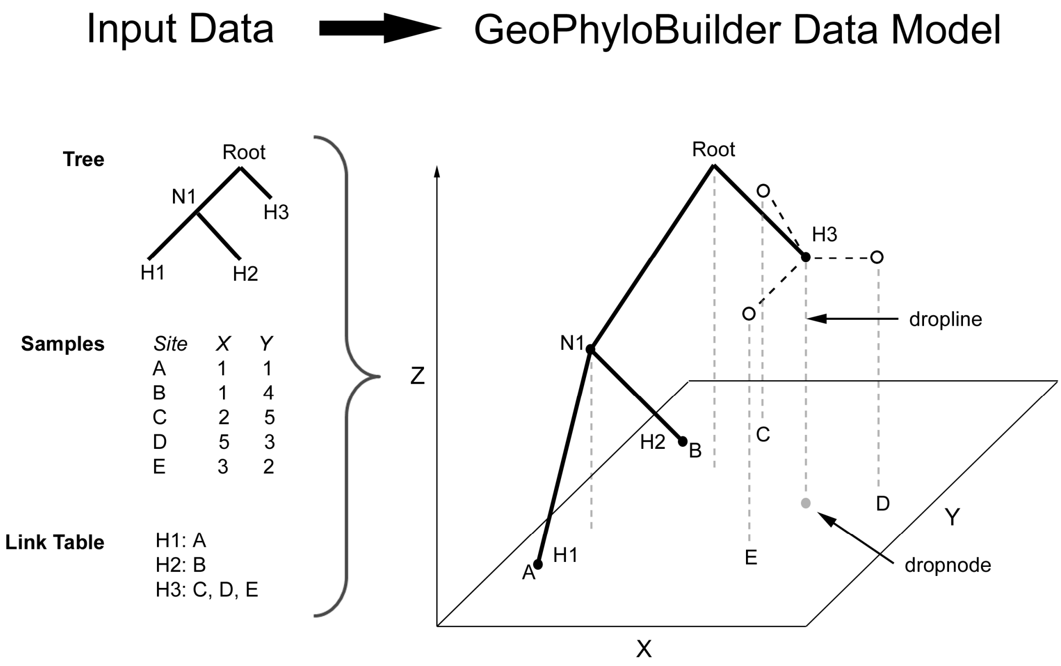
Figure 1. Example of Tree, sampling locations and link table.

GeoPhyloBuilder creates a GIS network model from a 'NEWICK' string defining the tree and an ArcGIS Geodatabase feature class, shapefile or .csv file (points only) defining the sample locations. An optional 'link table' defines n:m relationships between tree-tips and sample locations (fig. 2). Sample locations may be points, polylines or polygons.

NEWICK strings define trees through a set of nested parenthesise and ends with a semi-colon. Joe Felsenstein's provides additional information on the NEWICK format at http://evolution.genetics.washington.edu/phylip/newicktree.html.

The Brown Bear tree in fig. 1 is;

```
(((PyrA:1.83088,((CanB:0.641568,NorC:0.646183):0,Dal:0.582656):1.1597
8):0.736473,(((Abr:0.471139,Slo:0.441607):0,Cro:0.446221):0.980733,(G
re:0.779695,Bul:0.750162):0.685867):1.13363):4.67494,(((Rus:0.437607,
Est:0.408075):0,Ro2:0.412535):0.4373,Ro1:0.888444):6.37431);
```



| *Site* | *X* | *Y* |
|---|---|---|
| A | 1 | 1 |
| B | 1 | 4 |
| C | 2 | 5 |
| D | 5 | 3 |
| E | 3 | 2 |

H1: A
H2: B
H3: C, D, E

Figure 2. Creating a Geophylogeny.

The tree in fig. 2 is;

```
((H1,H2),H3);
```

To create a geophylogeny the position of phylogenetic nodes observed at multiple locations and those inferred from the data but not observed must be determined. GeoPhyloBuilder v1.1 implements four node positioning methods envelope centroid, mean centroid, minimum convex polygon centroid and direct vicariance analysis (DAVA) centroid. Tree distances or hierarchical levels are assigned to the z-attribute of the geophylogeny for simple 3D visualization in ArcScene. 'Droplines' are design elements that connect geophylogeny nodes to other geographical entities, e.g. a base map, positioned at other z-values. Trees can be re-rooted to any node or mid-point rooted.

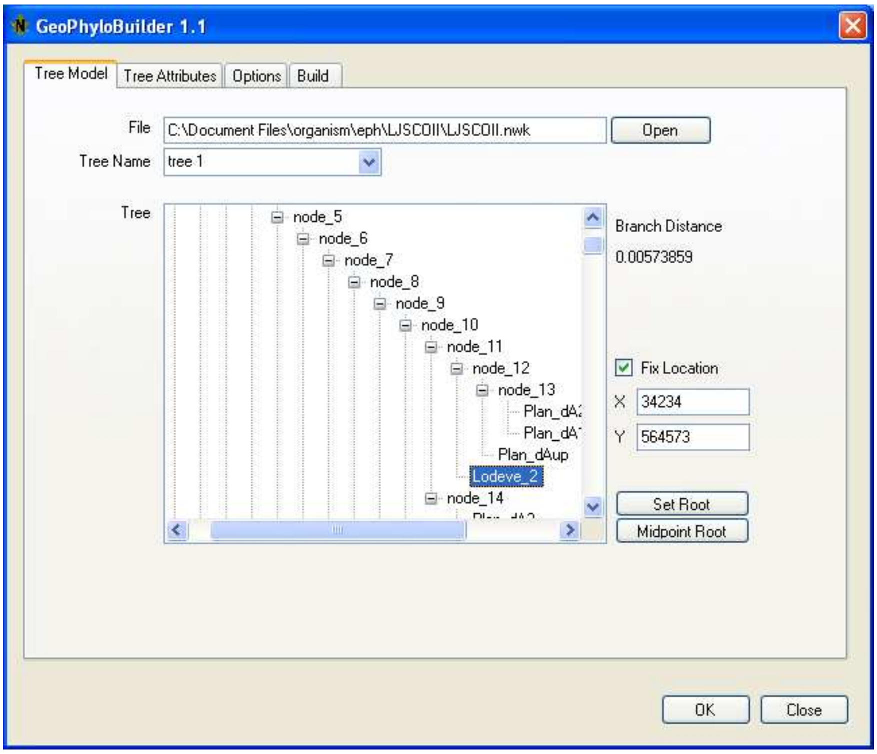
## Building Geophylogenies

### Running GeoPhyloBuilder

Start > All Programs > GeoPhyloBuilder > GeoPhyloBuilder.

### The Tree Model Tab

The Tree Model tab supports the selection of a tree model, re-rooting and the specification of fixed spatial coordinates for node positioning.



GeoPhyloBuilder 1.1

File C:\Document Files\organism\eph\LJSCOII\LJSCOII.nwk

Tree Name tree 1

Branch Distance 0.00573859

☑ Fix Location

X 34234

Y 564573

Figure 3. Tree Model tab.

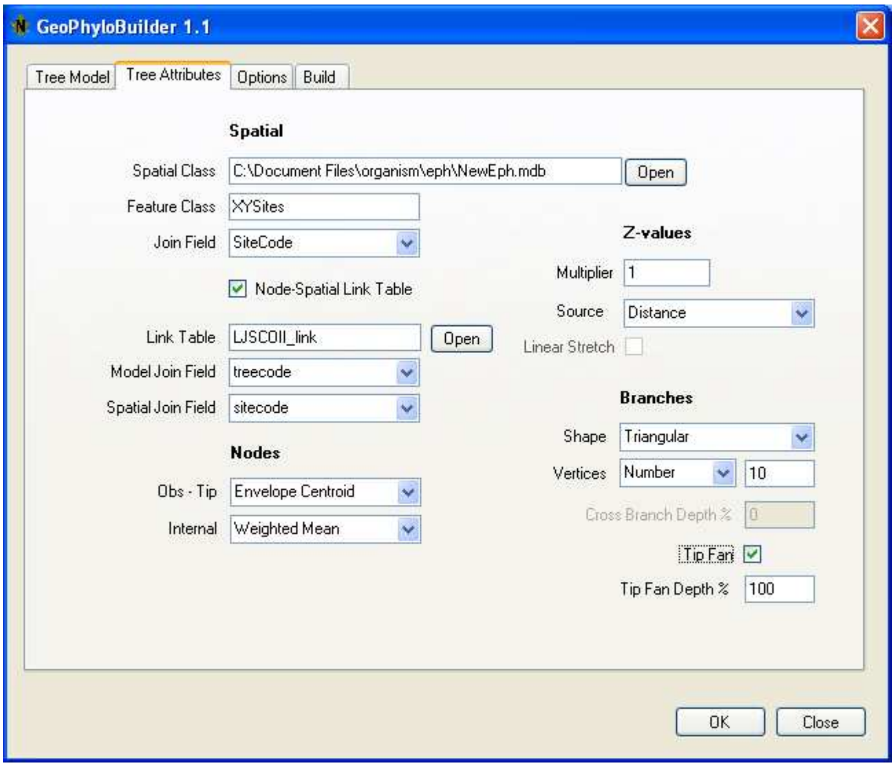
Select the ASCII file containing your NEWICK string. If more than one tree exists in the file select the desired model from the dropdown *Tree Name* menu. The selected tree is displayed in the tree window. To re-root an unrooted tree select the node to be designated the root and *Set Root*. The tree will be redrawn with the selected node set as the new root.

The *Midpoint Root* button will re-root the trees at the midpoint between the most distant nodes.

Nodes can be selected and their location specified using *Fix location*.

**Tree Attributes**

The Tree Attributes tab supports the selection of a spatial class defining the location of tip node entities (including n:m relationships between tips and locations), the algorithm to be used to position tips and internal nodes, and settings that determine tree depth and branch path.



Spatial

Spatial Class: C:\Document Files\organism\eph\NewEph.mdb

Feature Class: XYSites

Join Field: SiteCode

☑ Node-Spatial Link Table

Link Table: LJSCOII_link

Model Join Field: treecode

Spatial Join Field: sitecode

Nodes

Obs - Tip: Envelope Centroid

Internal: Weighted Mean

Z-values

Multiplier: 1

Source: Distance

Linear Stretch: □

Branches

Shape: Triangular

Vertices: Number 10

Cross Branch Depth %: 0

Tip Fan: ☑

Tip Fan Depth %: 100

Figure 4. Tree Attributes tab.

### *Spatial Source*

The *Data Source* specifies the location of data source that contains the feature class containing the spatial location of the tip node entities. Sources may be an ESRI Geodatabase or a directory containing one more shapefiles.

The *Feature Class* is the geodatabase feature class or shapefile within the data source that contains the feature class containing the spatial location of the tip node entities. Features may be point, polyline or polygon entities. As geophylogenies are built from points centroids are extracted from polygons and midpoints from polylines and the model built from these. If polygons are crescent-shaped the centroid may lie outside of the polygon extent, in such circumstances you may prefer to supply your own point shapefile.

The *Join Field* specifies the Feature Class field that contains the corresponding tree tip code. All tip-location combinations may be defined in the Feature Class or alternatively a link file specified which defines which tips are at each location specified in the Feature Class.

If a Link File is used then check the *Node –Spatial Link Table* checkbox and select the link table name. The link table may be an Access database table, Excel spreadsheet of comma separated text file. Select the *Model Join Field* to join the tree tip to the link table and the *Spatial Join Field* to join the link table to the Feature Class.

### *Node Positioning*

*Tip-Obs* sets the algorithm used to position tip node, while *Internal* sets the internal node algorithm. Two methods are implemented for spatial data in geographical and projected coordinate systems.  The *Envelope Centroid* method locates tip nodes at the centroid of the spatial envelope enclosing the associated observations and internal nodes at the centroid of nodes at the next lower level. The *Mean Position* method places nodes at the mean location of observations or the next lower level nodes.

Two addition methods, *MCP Centroid* and *DAVA Centroid,* are available for bifurcate trees with point observations in a projected coordinate system. DAVA stands for ‘Direct Area Vicariance Analysis’ and is inspired by Direct Vicariance Analysis

(Hovenkamp, 1997; Hovenkamp, 2001; Fattorini, 2007). If a tree has polytomies (more than two branches descending from a node) then the first two branches will be processed only.

The *MCP Centroid* option positions nodes at the centroid of a minimum convex polygon (MCP) of the set of observations defined by the node.

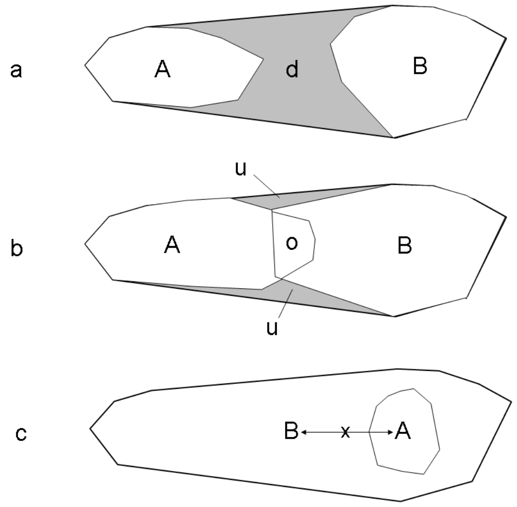


Figure 5. MCPs and DAVA positioning.

With DAVA if the daughter MCPs are disjunct then the node is placed at the centroid of the region of disjunction (d, 5a), however if they overlap but neither is enclosed by the other it is placed at the centroid of the overlap (o, 5b) and unoccupied space within the node MCP identified (u, 5b). In both MCP and DAVA, if one MCP encloses the other the node is by default placed at the midpoint between the MCP centroids (x, 5c). Alternatively it can be placed at the centroid of either the enclosing (A, 5c) or surrounding MCP (B, 5c).

A _mcp polygon feature class containing the node MCPs is output when the MCP or DAVA centroid methods are used. In addition, DAVA also outputs a _dava polygon class. Four parameters for MCP and DAVA including the enclosure positioning method can be set on the *Options* tab.

### *Branch Path*

Several parameters control how branches connect nodes (Fig 6). Triangular branches directly connect nodes with a line following the minimum distance. Rectangular branches follow the minimum distance spatial in the x-y plane then drop vertically to the daughter nodes. n Vertices assigns the number of vertices to place along each branch. Branches that contain vertices bend as a chain when projected into other coordinate systems. The greater the number of vertices the smoother the curve however the resulting model will be larger.

*Cross Branch Depth %* sets the position of rectangular cross branches in relation to the parent and daughter nodes.

Tip fans connect observations to a location between the 1st internal node and the location of the tip node as opposed to the tip node itself suggesting the separation of observations since the origin of the tip form.

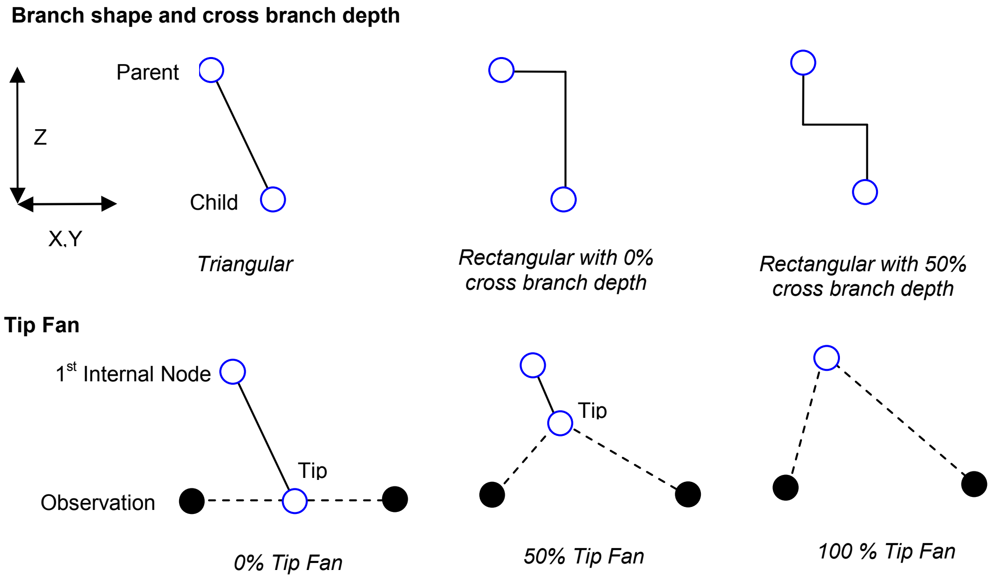

Branch shape and cross branch depth
Parent
Z
Child
X,Y
Triangular
Rectangular with 0% cross branch depth
Rectangular with 50% cross branch depth
Tip Fan
1st Internal Node
Tip
Observation
Tip
0% Tip Fan
50% Tip Fan
100 % Tip Fan

Figure 6. Branch Settings

### *Z Positioning*

The *Z Source* specifies whether tree depths should be derived from tree distance or the number of hierarchical levels from the root.

*Linear Stretch* scales the z-value of nodes in a cladogram with an uneven number of levels to create an ultrametric tree. Node z values are calculated as,

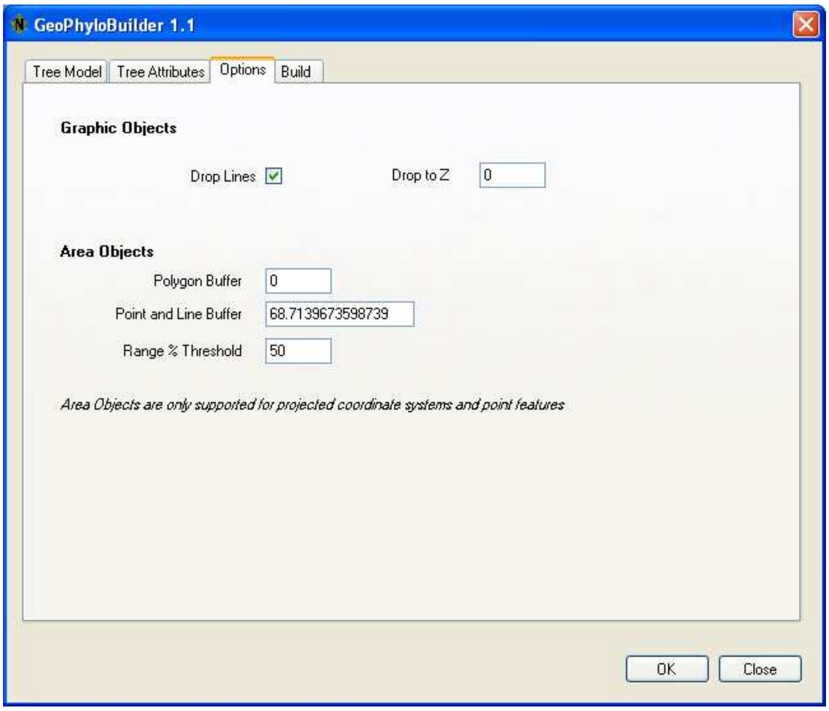
$$nodeZ = \left( nLtip / \left( \frac{nLroot}{nLtip} \right) * \max Ltree \right) * Zmult$$

where *nLtip* is the number of levels from the node is from the furthest tip it defines, *nLroot* is the number of levels the node is from the root, *max Ltree* is the deepest number of levels in the cladogram and *zMult* is the z -multiplier constant.

The z Multiplier multiplies tree distances by the value specified to aid 3D visualization. For example, if an entire tree has a depth of 1 unit is displayed on a map with meters as the base unit then the tree will by default only be raised to 1 meter above the ground surface, i.e. not very far. Note that a z-elevation multiplier can be also be set in ArcScene on the 'Base Heights' tab of a features properties.

**Options tab**

The Options tab accesses graphic and area object settings (Fig 7).



**Graphic Objects**

Drop Lines ☑ Drop to Z 0

**Area Objects**

Polygon Buffer 0

Point and Line Buffer 68.7139673598739

Range % Threshold 50

*Area Objects are only supported for projected coordinate systems and point features*

Figure 7. Options Tab

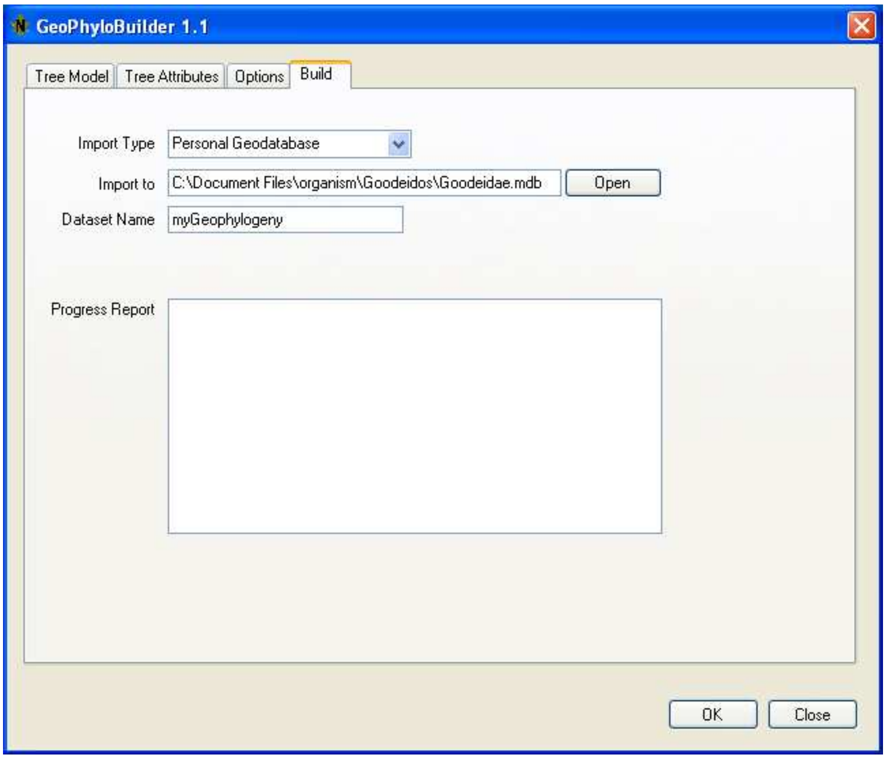
Selecting *Drop Lines* will create a feature class containing a set of vertical lines from the nodes to a base elevation. Change *Drop to Z* for lines to be dropped to a non-zero elevation.

*MCP and DAVA* options may also be set on the *Options* tab. *Polygon Buffer* buffers MCPs by a set distance. As MCPs cannot be built for single or pairs of observations they are converted to polygons by applying a buffer around a single point or the line connecting a pair of points. The width of this buffer is the *Point and Line Buffer* (default = 1/10,000th of the longest dimension of the spatial envelope).

DAVA calculates the % overlap between child MCPs. The *DAVA Threshold* classifies overlapping DAVA polygons by the maximum percentile overlap of the child MCPs. These classes are stored in the _dava PolyType field (see feature class definitions).

### Build tab

The *Build* tab (Fig 8) supports the selection of output to a geodatabase set or file folder which will contain a folder of shapefiles. The *Dataset Name* is either the name of the geodatabase feature dataset that will contain output feature classes or a folder that will contain output shapefiles.



Import Type: Personal Geodatabase

Import to: C:\Document Files\organism\Goodeidos\Goodeidae.mdb

Dataset Name: myGeophylogeny

Figure 8. Build Tab.

## GeoPhyloBuilder Output

Depending on the settings two or more feature classes are output to the selected geodatabase dataset or shapefile folder. Feature classes are prefixed with the input dataset name and suffixed with '_node', '_branch', '_dropline', '_mcp' and '_dava' to their contents.

### _node

| Idx | Field Name | Type | Description |
|---|---|---|---|
| 0 | FID | Integer | Internal Feature Identifier |
| 1 | Shape | Point | Geometry |
| 2 | NodeID | Long Integer | Geophylobuilder defined unique node identifier |
| 3 | NodeName | String(30) | Name of node |

|  |  |  |  |
|---|---|---|---|
| 4 | NodeType | Integer | The type of node;<br>0 = root<br>1 = tip<br>2 = internal not root<br>3 = observation |
| 5 | nLRoot | Integer | Number of level from root where root is level 0 |
| 6 | nLTip | Integer | Number of levels Level to deepest tip defined by the node. |
| 7 | d | Double | Tree distance from parent node. |
| 8 | dRoot | Double | Tree distance from root. |
| 9 | dTip | Double | Tree distance to deepest tip defined by node. |
| 10 | DisplayZ | Double | Z-value of node. |
| 11 | RankID | String (254) | Node rank; root 0; first level nodes 0.0, 0.1, 0.2, etc; second level nodes 0.0.0,0.0.1,0.1.0, etc |

## _branch

| Idx | Field Name | Type | Description |
|---|---|---|---|
| 0 | FID | Integer | Internal Feature Identifier |
| 1 | Shape | Polyline | Geometry |
| 2 | BranchID | Integer | Geophylobuilder defined unique branch identifier |
| 3 | BranchName | String(30) | Name of branch |
| 4 | pNodeID | Integer | Parent NodeID |
| 5 | cNodeID | Integer | Child NodeID |
| 6 | BranchType | Integer | The type of branch;<br>1 = tree branch<br>2 = tip-observation branch |
| 7 | PhyloLength | Double | Branch depth |
| 8 | GeoLength | Double | Geographic length of branch |
| 9 | pRankID | String(254) | RankID of parent node |

| | | | |
|---|---|---|---|
| 10 | cRankId | String(254) | RankID of child |

## _dropline

| Idx | Field Name | Type | Description |
|---|---|---|---|
| 0 | FID | Integer | Internal Feature Identifier |
| 1 | Shape | Polyline | Geometry |
| 2 | DLineID | Integer | Geophylobuilder defined unique dropline identifier |
| 3 | DLineName | String(30) | Name of DropLine |
| 4 | DLineType | Integer | The type of node the dropline descends from;<br>0 = root<br>1 = tip<br>2 = internal not root<br>3 = observation |
| 5 | NodeID | Integer | NodeID from Node FClass |

## _mcp

| Idx | Field Name | Type | Description |
|---|---|---|---|
| 0 | FID | Integer | Internal Feature Identifier |
| 1 | Shape | Polygon | Geometry |
| 2 | PolyID | Integer | Geophylobuilder defined unique polygon identifier |
| 3 | NodeID | Integer | NodeID from Node |
| 4 | NodeName | String | NodeName from Node |

## _dava

| Idx | Field Name | Type | Description |
|---|---|---|---|
| 0 | FID | Integer | Internal Feature Identifier |
| 1 | Shape | Polygon | Geometry |
| 2 | PolyID | Integer | Geophylobuilder defined unique polygon identifier |

| | | | |
|---|---|---|---|
| 3 | NodeID | Integer | NodeID from node |
| 4 | NodeName | String | NodeName from Node |
| 5 | cNodeID | Integer | Child NodeID from node |
| 6 | cNodeName | String | Child NodeName from Node |
| 7 | PolyType | Integer | The type of polygon.<br>0 = Disjunction<br>1 = Overlap below DAVA threshold<br>2 = Unoccupied space where overlap is below DAVA threshold<br>3 = Overlap above DAVA threshold<br>4 = Unoccupied space where overlap is above DAVA threshold<br>5 = Non-overlapping region of child MCP |
| 8 | Overlap | Double | % Area overlap |

## Defining Subclades

RankId's provide a quick way to select and code subclades through 'path enumeration'. Each node in a tree is coded according to the following schema. The root is designated 0. All nodes at the level below the root are then coded 0.0, 0.1, 0.2, etc. Subsequent divisions are coded through the addition of an extra dot and number hence, 0.0.1, 0.0.2 and 0.0.3 are three nodes below 0.0, and 0.2.0 and 0.2.1 are two nodes below 0.2 (Fig 9).

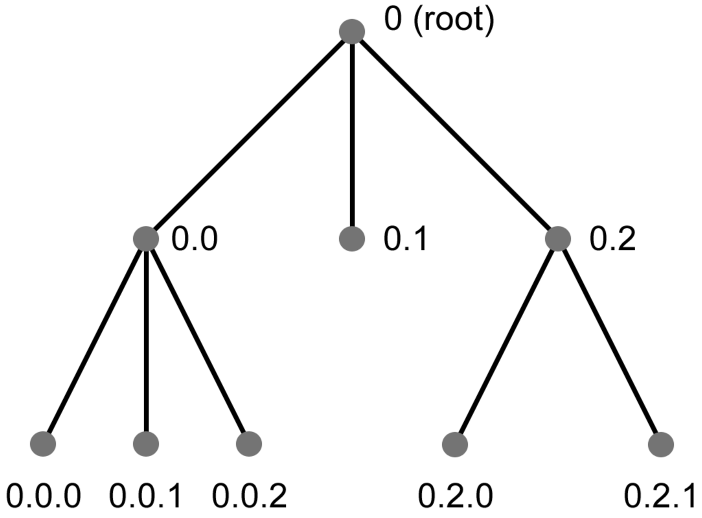


Figure 9 Node RankId.

To code subclades create a new field in the node and branch feature classes, e.g. 'mysubclade'. Select subclade nodes by 'selecting by attributes' where the select string is of the form *SELECT * FROM tree.tree_node WHERE rankid like '0.2*'* that will select the node with rankid = 0.2 and all nodes below it. The 'mysubclade' field can now be updated to a code describing the subclade group, e.g. 'mysubclade1'. A similar logic can be applied to select the branches of subclades using the FromRankID or ToRankID fields.

## References

Fattorini S. Hovenkamp’s ostracized vicariance analysis: testing new methods of historical biogeography. Cladistics (2007) 23: 1-12 (early online).

Hovenkamp P. Vicariance events, not areas, should be used in biogeographical analysis. Cladistics (1997) 13: 67-78.

Hovenkamp P. A direct method for the analysis of vicariance patterns. Cladistics (2001) 17: 260-265.

Kidd DM, Liu X. GEOPHYLOBUILDER 1.0: an ArcGIS extension for creating 'geophylogenies'. Mol. Ecol. Resour. (2008) 8: 88-91.