PHYLO3D MANUAL

1. Overview

phylo3D is a set of packages for converting phylogenetic trees in New Hampshire (also called Newick or Phylip) format, New Hampshire Extended, NEXUS and the NCBI taxonomy database into LibSea format.

LibSea format can be read by Walrus a java-based graph visualization tool that can visualise graphs with several thousand nodes in 3D hyperbolic space. Thus, phylo3D makes it possible to visualise "huge" phylogenetic trees in 3D hyperbolic space.

The advantage of visualising huge trees in this way is that it provides a focus+context view of the phylogenetic tree.

2. Obtaining and installing

Phylo3D is freely downloadable (http://www.ii.uib.no/~tim/).

Walrus was developed by Young Hyun at the Cooperative Association for Internet Data Analysis (CAIDA) and is available in binary form upon request from CAIDA (http://www.caida.org/tools/visualization/walrus).

For details of how to install phylo3D and Walrus, refer to the file /doc/INSTALL_NOTES.txt in the phylo3D distribution.

3. Formats

The user does not need to have a detailed knowledge of the formats since it is the purpose of the phylo3D to carry out the conversions. However, for the purpose of completeness links to format descriptions are included here.

There are four formats involved in phylo3D conversions:

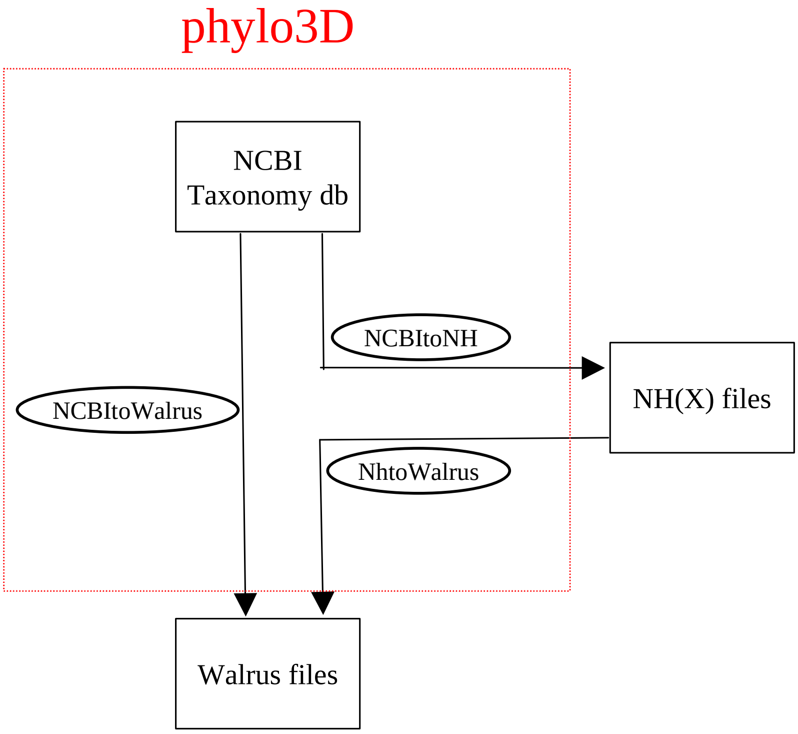
- The NCBI taxonomy database flat file data and format description
  - downloadable from: ftp://ftp.ncbi.nih.gov/pub/taxonomy/
  - this data has been downloaded from NCBI and is part of phylo3D as a serialized object. The user therefore does not need to downloaded or relate to this format. The data incorporated into phylo3D were downloaded in September 2003.
- The New Hampshire format (.nh) also called Newick or Phylip
  - This is the most common format for phylogenetic trees
  - A description of this format is available from: http://evolution.genetics.washington.edu/phylip/newicktree.html
- The New Hampshire Extended format (.nhx)
  - This format is an extended version of the NH format. It introduces tags to associate various data fields with nodes in a phylogenetic tree
  - A description of this format is available from: http://www.genetics.wustl.edu/eddy/forester/NHX.pdf
- The Libsea format

- This is the format by the Walrus tree visualization tool
- A description of the format is available within the CAIDA Walrus distribution (docs/graph-format.txt)

4. Using the tools

The phylo3D package contains the data from the NCBI taxonomy database and 3 conversion tools:

1. NCBItoWalrus: converts data from the NCBI taxonomy database to LibSea format (Walrus)
2. NCBItoNH: converts data from the NCBI taxonomy database to NH or NHX format
3. NHtoWalrus: converts from NH, NHX or NEXUS to LibSea format (Walrus)



NCBItoWalrus

The user specifies which subtree of the NCBI taxonomy database is to be included in the output by specifying the taxonomy ID of the root node of the subtree of interest. The user can also specify a colouring of this subtree by specify pairs of NCBI taxonomy IDs and RGB colour codes in hexadecimal format. All nodes in the subtree rooted at the node with the specified taxonomy ID are coloured with the specified colour. The NCBI taxonomy IDs can be retrieved from the NCBI taxonomy database (http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html).

Note that the colouring is carried out in the specified order so that if the user wishes to colour a subtree a particular colour then the root node of the supertree should be specified first followed by the root node of the subtree. If the opposite order is used then the supertree colour will overwrite the subtree colour. If the user colours the tree, then it is also necessary to specify whether the branch leading to the node which is the root of subtree should be coloured. The advantage of colouring the branch is that it makes the subtree more visible. This is especially useful if the tree is large and the subtree of interest is close to the leaves of the tree.

The options must appear in the following order:
--narrowTo=node (obligatory) where node is an NCBI taxonomy ID
--colours=node,colour+node,colour+node,colour (optional) where node is NCBI taxonomy ID and colour is an RGB colour in Hex format
--colourSubtreeRootBranch=Y or N (obligatory if the 'colours' option is used)
An output file must also be specified (e.g. output.graph)

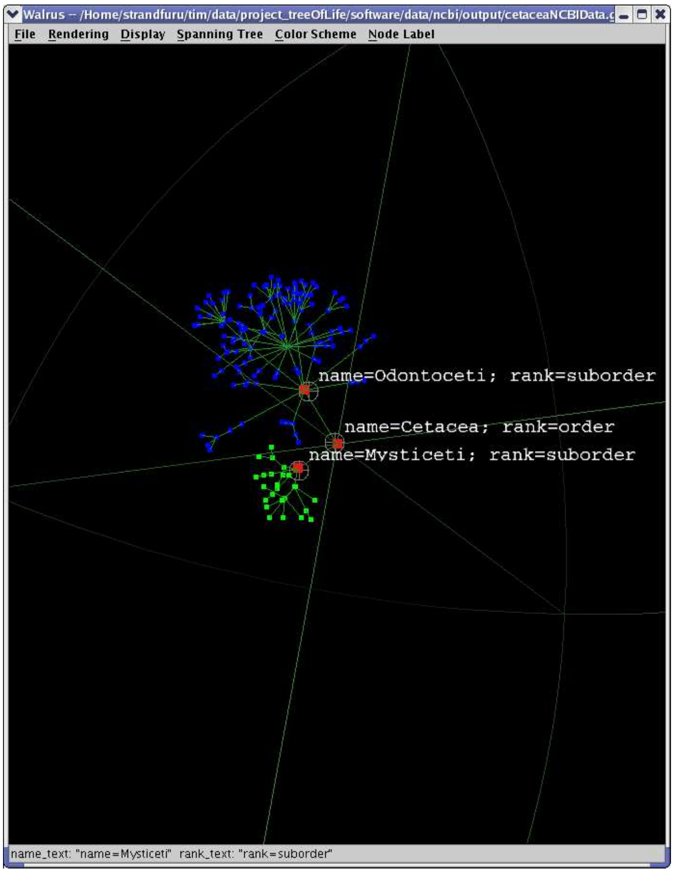
For example, the command line for producing the cetacea.graph sample (see doc/samples/cetecea.graph) is:
In Windows (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar;ATVapp.jar;. phylo3D.tools.NCBItoWalrus --
narrowTo=9721 --colours=9761,00ff00+9722,0000ff --colourSubtreeRootBranch=N
cetacea.graph
```

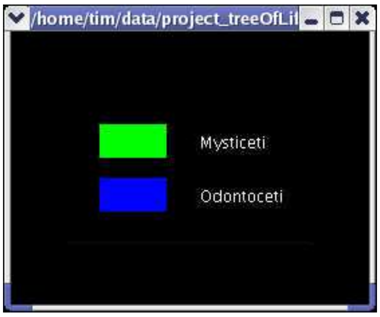
In Unix (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar:ATVapp.jar:. phylo3D.tools.NCBItoWalrus --
narrowTo=9721 --colours=9761,00ff00+9722,0000ff --colourSubtreeRootBranch=N
cetacea.graph
```

The result of the above command line will be the creation of the file “cetecea.graph” in Walrus format (which will contain the subtree of the NCBI taxonomy database rooted at node 9721) and the creation of a legend file in JPEG format (cetecea.jpg) corresponding to the colouring specified by the user. This file can be opened in a JPEG viewer and used as extra visual information while navigating the tree in Walrus.


Walrus – /Home/strandfuru/tim/data/project_treeOfLife/software/data/ncbi/output/cetaceaNCBIData.g
File
Rendering
Display
Spanning Tree
Color Scheme
Node Label
name=Odontoceti; rank=suborder
name=Cetacea; rank=order
name=Mysticeti; rank=suborder

name_text: "name=Mysticeti" rank_text: "rank=suborder"



🟩 Mysticeti
🟦 Odontoceti

NCBItoNH

The user specifies which subtree of the NCBI taxonomy database is to be included in the output by specifying the taxonomy ID of the root node of the subtree of interest. The user must also specify whether the sequence name used in the output should be the NCBI taxonomy IDs or the scientific species names, whether internal node data is to be included in the output and what output format is to be used.

The options must appear in the following order:
--narrowTo=node where node is NCBI taxonomy ID (obligatory)
--sequenceName=T or S, where T: NCBI taxonomy IDs and S: NCBI scientific species names (obligatory)
--internalNodes=Y or N, where Y:internal node data is included in output and N: data not included (obligatory)
--outputFormat=NH or NHX (obligatory)
An output format must also be specified (e,g, output.nhx)

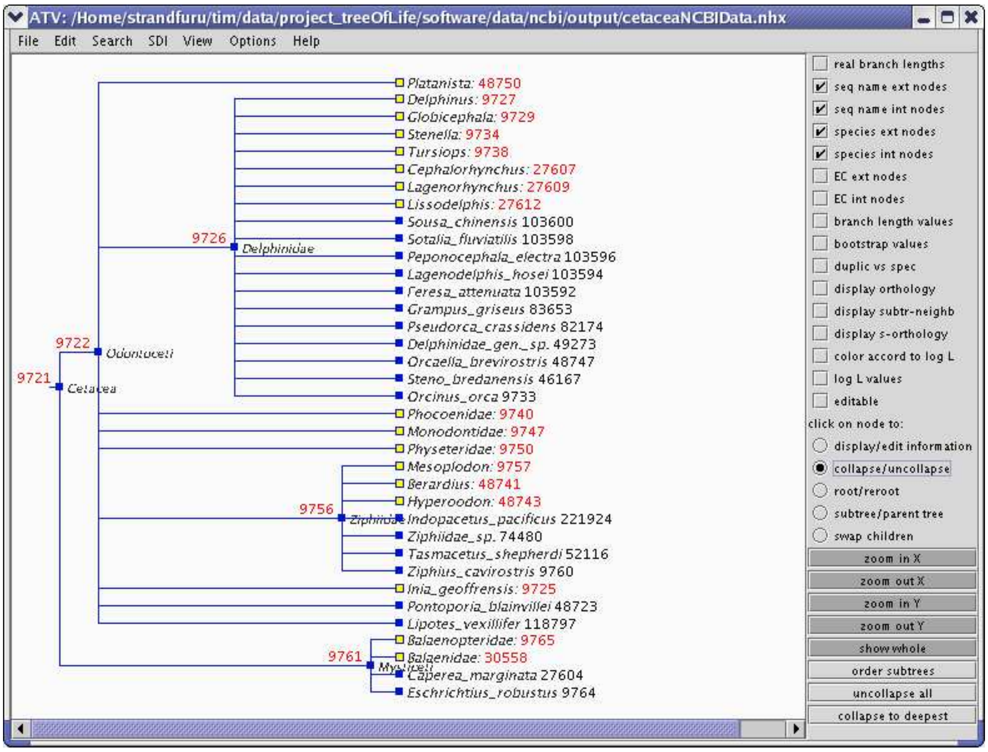
The command line for producing the cetacea.nhx sample (see docs/samples/cetacea.nhx) is:
In Windows (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar;ATVapp.jar;. phylo3D.tools.NCBItoNH --narrowTo=9721
--sequenceName=T --internalNodes=Y--outputFormat=NHX cetacea.nhx
```

In Linux (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar:ATVapp.jar:. phylo3D.tools.NCBItoNH --narrowTo=9721
--sequenceName=T --internalNodes=Y--outputFormat=NHX cetacea.nhx
```

The result of the above command line is the creation of the output file specified by the user (cetecea.nhx) which will contain the specified subtree in NHX format, where the sequence name are the taxonomy IDs and the internal node data is included. This tree can be visualized in for example the phylogenetic tree visualization tools phylip or ATV (as illustrated below).



ATV: /Home/strandfuru/tim/data/project_treeOfLife/software/data/ncbi/output/cetaceaNCBIData.nhx

□ real branch lengths
☑ seq name ext nodes
☑ seq name int nodes
☑ species ext nodes
☑ species int nodes
□ EC ext nodes
□ EC int nodes
□ branch length values
□ bootstrap values
□ duplic vs spec
□ display orthology
□ display subtr-neighb
□ display s-orthology
□ color accord to log L
□ log L values
□ editable

click on node to:
○ display/edit information
● collapse/uncollapse
○ root/reroot
○ subtree/parent tree
○ swap children

NHtoWalrus

The user must specify the format of the input file (NH, NHX or NEXUS). For an example of the tree block in the NEXUS format that can be parsed see docs/samples/nexusTreeBlock.nex The user can also specify a colouring of the tree by specify pairs of sequence names (this is the text representing each node in the input tree) and RGB colour codes in hexadecimal format. All nodes in the subtree rooted at the node with the specified taxonomy ID are coloured with the specified colour.

Note that the colouring is carried out in the specified order so that if the user wishes to colour a subtree a particular colour then the root node of the supertree should be specified first followed by the root node of the subtree. If the opposite order is used then the supertree colour will overwrite the subtree colour. If the user colours the tree, then it is also necessary to specify whether the branch leading to the node which is the root of subtree should be coloured. The advantage of colouring the branch is that it makes the subtree more visible. This is especially useful if the tree is large and the subtree of interest is close to the leaves of the tree.

In addition it is possible to add taxonomic information to the input tree. This is useful in itself but can also be used for colouring the tree and thus giving an indication of the extent to which the tree agrees with the taxonomy. This taxonomic information must be provided in an input file. This file must be in plain text format. Each line must be a tab (or space) delimited taxonomic classification. Each classification must have the same number of levels. The user then also specifies a matching column which is the column containing strings that will be used to search the sequence names of the leaf nodes in the tree file. When the program finds a match, it uses the match to give the matching node a taxonomic classification. The user also specifies a colouring column which is the taxonomic level at which the leaf nodes of the tree will be taxonomically classified and coloured (see further down for a concrete example). If the colouringNames option is not used the program will then colour the Walrus tree according to the taxonomy level chosen through the colouring column. If the colouringNames option is used the user must specify, a set of colouringStrings and colours, where the colouringStrings are strings that appear in the colouring column, in this case only the leaf nodes with the colouringString classification will be coloured.

Note that it is possible to combine the colouring of specific subtrees(the –colours option) with the taxonomic colouring (--taxonomyFile, --matchingColumn, --colouringColumn, --colouringNames options)

The options must appear in the following order:
--inputFormat=NH, NHX or NEXUS (obligatory) specifies the format of the input tree
--colours=node,colour+node,colour+node,colour (optional) where node is a sequence name in the input tree and colour is an RGB colour in Hex format. Enables the colouring of a subtree rooted at a specific node
--taxonomyFile=fileName (optional) where fileName is the path to the file containing the taxonomy
--matchingColumn=digit (obligatory if taxonomyFile option used
where digit is the number of the column to be used for matching (first column is 1)
--colouringColumn=digit (obligatory if taxonomyFile option used)
where digit is the number of the column to be used for colouring (first column is 1)
--colouringNames=colouringString,colour+colouringString,colour (optional)
where colouringString is a string from the colouring column
where colour is the hex representation of an RGB code

--colourSubtreeRootBranches=Y or N (default is N, obligatory if colouring either specific nodes or colouring using a taxonomy (or both)

An input file (e.g. input.nhx) and an output file (e.g. output.graph) must also be specified.

The command line for producing the cetacea in LibSea format (docs/samples/cetacea.graph) from the cetacea data in NHX format (docs/samples/cetacea.nhx) would be:
In Windows (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar;ATVapp.jar;. phylo3D.tools.NHtoWalrus --
inputFormat=NHX --colours=9761,00ff00+9722,0000ff --colourSubtreeRootBranch=Y
./doc/samples/cetaceaNCBIData.nhx cetacea.graph
```

In Linux (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar:ATVapp.jar:. phylo3D.tools.NHtoWalrus --
inputFormat=NHX --colours=9761,00ff00+9722,0000ff --colourSubtreeRootBranch=Y
./doc/samples/cetaceaNCBIData.nhx cetacea.graph
```

The result of the above command line will be the creation of the output file specified by the user (cetacea.graph which will contain the input file data in walrus format) and the creation of a legend file containing a legend in JPEG format corresponding to the colouring specified by the user (cetacea.jpg). This file can be opened in a JPEG viewer and used as extra visual information while navigating the tree in Walrus (see NCBItoWalrus).

The “simple” files (docs/samples/simple.*) are the result of colouring a tree using a taxonomy. The command line would be:
In Windows (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar;ATVapp.jar;. phylo3D.tools.NHtoWalrus --
inputFormat=NHX --taxonomyFile=./doc/samples/taxo.txt --matchingColumn=1 --
colouringColumn=3 --colourSubtreeRootBranch=Y ./doc/samples/tree.nhx simple.graph
```

In Linux (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar:ATVapp.jar:. phylo3D.tools.NHtoWalrus --
inputFormat=NHX --taxonomyFile=./doc/samples/taxo.txt --matchingColumn=1 --
colouringColumn=3 --colourSubtreeRootBranch=Y ./doc/samples/tree.nhx simple.graph
```

The result of the above command line will be the creation of the output file specified by the user (simple.graph which will contain the input file in walrus format) and the creation of a taxonomy legend file in JPEG format with a different colour for each different taxonomic group at the taxonomic level specified by the colouringColumn (simpleTaxonomyLegend.jpg).

The “complex” files (/docs/samples/complex.*) are the result of colouring a tree using a taxonomy, but instead of colouring all the nodes according to the taxonomy, specific taxonomic groups are coloured (the --colouringNames option) and in addition some specific nodes are chosen for colouring (the –colours option).
In Windows (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar;ATVapp.jar;. phylo3D.tools.NHtoWalrus --
inputFormat=NHX --colours=nine,ff0000 –taxonomyFile=./doc/samples/taxo.txt --
matchingColumn=1 --colouringColumn=3 --colouringNames=canis,00ff00+fish,0000ff --
colourSubtreeRootBranch=Y ./doc/samples/tree.nhx complex.graph
```

In Linux (and assuming the current directory is where the jar files are located):

```
java –Xmx126M –cp phylo3D.jar:ATVapp.jar:. phylo3D.tools.NHtoWalrus --
inputFormat=NHX --colours=nine,ff0000 –taxonomyFile=./doc/samples/taxo.txt --
matchingColumn=1 --colouringColumn=3 --colouringNames=canis,00ff00+fish,0000ff --
colourSubtreeRootBranch=Y ./doc/samples/tree.nhx complex.graph
```

The result of the above command line will be the creation of the output file specified by the user (complex.graph) which will contain the input tree (tree.nhx) in walrus format and the creation of a taxonomy legend file in JPEG format (complexTaxonomyLegend.jpg) with a different colour for each taxonomic group specified in the –colouringNames option. Note that the colouring names “canis” and “fish” are in the third column of the taxo.txt file. In addition, the node with sequence name “nine” will be coloured red as specified by the –colours option and the details of this colouring will be output in the specific colouring legend file (complexLegend.jpg).

5. Samples

There are a number of samples of output from phylo3D in the directory doc/samples.

6. Common problems

If the tools throw OutOfMemoryExceptions then you should add the -Xmx option to your command line to increase the maximum amount of memory the JDK can use. To increase this to 500M (when using for example NCBItoWalrus), you would use the following command line:

In Windows (and assuming the current directory is where the jar files are located):

```
java -Xmx500M –cp phylo3D.jar;ATVapp.jar;. phylo3D.tools.NCBItoWalrus --
narrowTo=9721 --colours=9761,00ff00+9722,0000ff --colourSubtreeRootBranch=N
cetacea.graph
```

In Unix (and assuming the current directory is where the jar files are located):

```
java -Xmx500M –cp phylo3D.jar:ATVapp.jar:. phylo3D.tools.NCBItoWalrus --
narrowTo=9721 --colours=9761,00ff00+9722,0000ff --colourSubtreeRootBranch=N
cetacea.graph
```

7. References

The NCBI taxonomy database
Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Rapp BA, Wheeler DL (2000). GenBank. Nucleic Acids Res 2000 Jan 1;28(1):15-18

The Walrus tool:
http://www.caida.org/tools/visualization/walrus

The forester package which contains some classes used by phylo3D and the ATV tree viewer:
Zmasek C.M. and Eddy S.R. (2001) ATV: display and manipulation of annotated phylogenetic trees.Bioinformatics, 17, 383-384.

8. Contact

Reporting bugs, comments, suggestions relating to phylo3D: please email tim@cbu.uib.no