Overview of Rich Newick Strings

16 February, 2012

The purpose of Rich Newick strings is to succinctly represent the main features of phylogenetic networks in a character encoded human readable form. The format is inspired primarily by the Newick format for phylogenetic tree representation and the Extended Newick format for phylogenetic network representation as proposed by Cardona et al [1].

For the purposes of Rich Newick we define a phylogenetic network to be either:

- a unrooted phylogenetic X-tree[1]
  or
- a connected phylogenetic X-network (defined below)

used to model evolutionary relationships.

We define a connected phylogenetic X-network N to be an ordered pair (G, f), where

- G = (V, E) is a connected, directed, acyclic graph (connected DAG) with V = {r}, V_L V_T V_N, where:
  – indeg(r) = 0 (r is the root of N);
  – for all v an element of V_L, indeg(v) = 1 and outdeg(v) = 0 (V_L are the leaves of N);
  – for all v an element of V_T, indeg(v) = 1 and outdeg(v) >= 1 (V_T are the tree-nodes of N); and,
  – for all v an element of V_h, indeg(v) >= 2 and outdeg(v) = 1 (V_h are the hybrid nodes of N).
- f : V_L -> X is the leaf-labeling function, which is a bijection from $V_L$to X .

1. Representing Phylogenetic X-Networks

Consider the following phylogenetic X-network with X = { 1, 2 }:

[1] see [2] p.17

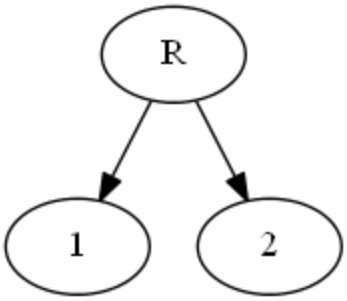

Figure 1.1

To encode the example phylogenetic X-network above in Rich Newick we can begin by representing the root of the network with the string “R;”. In this case the use of the character “R” corresponds to the label *R* in our network and the use of the “;” character terminates the Rich Newick string. (Note that the label *R* is a convenience label and not a member of the label set X in the X-network itself.) We can then encode the nodes incident to *R*, namely *1* and *2*, by prepending a comma separated list of *R*’s direct successors enclosed by parentheses to the character “R”:

(1, 2)R;

The ordering of *R*’s direct successors in the string is arbitrary. We could just as easily have encoded the phylogenetic X-network as:

(2, 1)R;

In general for Rich Newick strings if there is more than one possible way to encode a phylogenetic network any encoding is acceptable. However, tools that consume Rich Newick strings might use the orderings of successors within the string as hints for layout if the tool produces visualizations; as such it is advisable to sequence successors in a fashion that preserves the ordering of motivating figures or diagrams if applicable.

For encoding phylogenetic X-networks, the relationship between a given node *W* and its prepended node list *(Y1, Y2, Y3, …, Yn)* such as:

(Y1, Y2, Y3)W;

is that there exist edges connecting the given node (the source) to the nodes within its prepended node list (the destinations). In our example this would be the edges (*W*, *Y1*), (*W, Y2*) and (*W, Y3*) respectively.

The Rich Newick format is largely white-space agnostic. We define whitespace to be the characters: space, tab (\t), carriage return (\r) and newline (\n). Whitespace may

be interleaved at any position within a Rich Newick string without changing the string's semantics *with the exception* of within node labels and edge attributes (both discussed in later sections). So, for example, the Rich Newick string:

(1,2)R;

has the same meaning as the string:

(1 , 2) R;

or even the string:

(1
,
2
)R;

To encode networks of increasing complexity we can recursively apply the *(..., ..., ...)...* notation in a nested fashion. For example, the following phylogenetic X-network with X = { 1, 2, 3, 4, 5, 6, 7} :

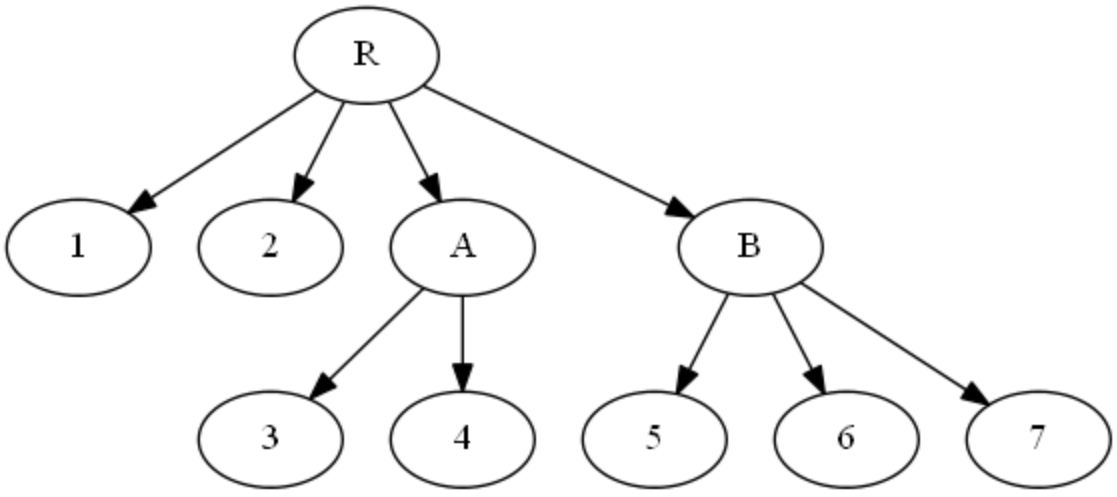

Figure 1.2

...can be encoded as:

(1, 2, (3, 4)A, (5, 6, 7)B)R;

where each inner parenthetic list element (i.e. *(3, 4)A* and (*5, 6, 7)B*)) represents a

subnetwork in the given phylogenetic X-network. By allowing any depth of nesting of the node lists within other node lists (e.g. *(3, 4)* prepending *A*--itself a node in the list of *R*) we can represent networks of any depth.

2. Representing Unrooted Phylogenetic X-Trees.

To declare that a string encodes a unrooted phylogenetic X-tree instead of a phylogenetic X-network we prepend the Rich Newick string with the "[&U]" or "[&u]" characters (the choice is arbitrary). For example the unrooted phylogenetic X-tree with X = {1, 2, 3, 4, 5, 6}:

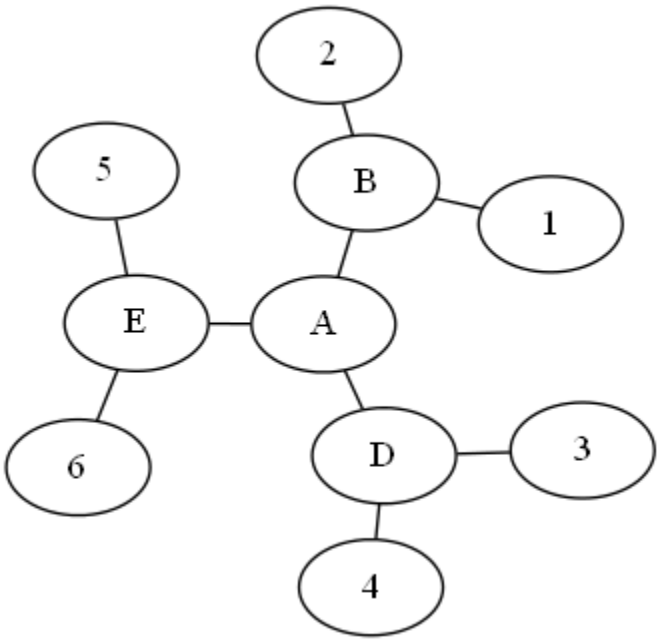

Figure 2.1

… can be encoded as:

[&U]((1, 2)B, (3, 4)D, (5, 6)E)A;

Encoding unrooted phylogenetic X-trees follows a similar form to that of phylogenetic X-networks with one key difference: if the outermost (and *only* the outermost) node list contains exactly two nodes (irrespective of whether these nodes are themselves prepended with node lists) then the outer node list is interpreted differently from the convention established for phylogenetic X-networks. Namely in this case for unrooted phylogenetic X-trees the two nodes in the outermost node list are considered to be adjacent to each other. Further, it is considered an error to append a node label to the outermost node list in this case.

For example, the Rich Newick string:

[&U]((1, 2)A, (3, 4)B);

represents a unrooted phylogenetic X-tree that could be diagrammed as:

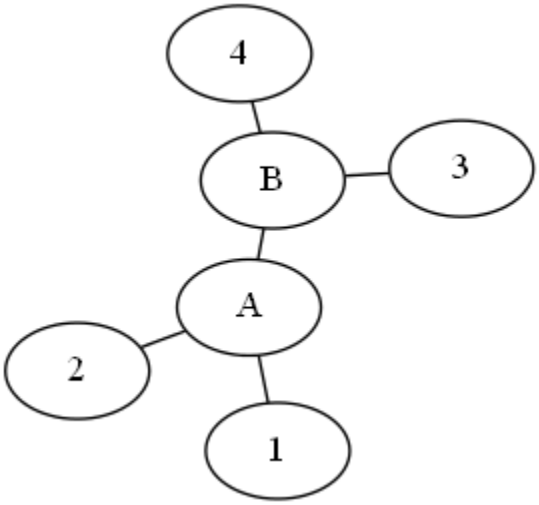

Figure 2.2

Note how *A* and *B* are considered adjacent even though they appear in the same node list. This would not be the case for a phylogenetic X-network. Nor would this be the case for a unrooted phylogenetic X-tree with an outer node list of three nodes. Only because the encoded network is a unrooted phylogenetic X-tree with two nodes in the outer node list do we consider those two nodes (in our example *A* and *B*) as adjacent. This is done so Rich Newick strings can conform to a corresponding historical convention as established by the format of unrooted Newick strings.

It should be noted that we may also, in similar fashion to unrooted X-trees, explicitly declare that a network be interpreted as a phylogenetic X-network by use of the "[&R]" or "[&r]" characters; however, this is the default interpretation of networks even when no such characters are included.

3. More on Node Labels

Node labels are used to identify nodes within a network. For example in the phylogenetic X-network with X = {1, 2}:

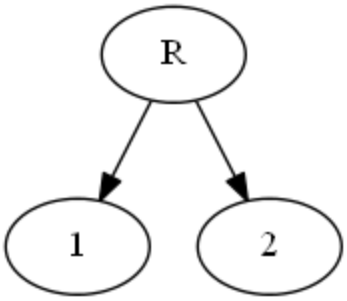

Figure 3.1

graph nodes *R*, *1* and *2* can be labeled within the Rich Newick string using the “R”, “1” and “2” characters:

(1, 2)R;

However, node labels are optional for nodes of outdegree ≠ 0 in phylogenetic X-networks and degree > 1 in phylogenetic X-trees (i.e. the non-leaf nodes). As such we could encode the phylogenetic X-network above (omitting the R label) simply as:

(1, 2);

For further example the phylogenetic X-network:

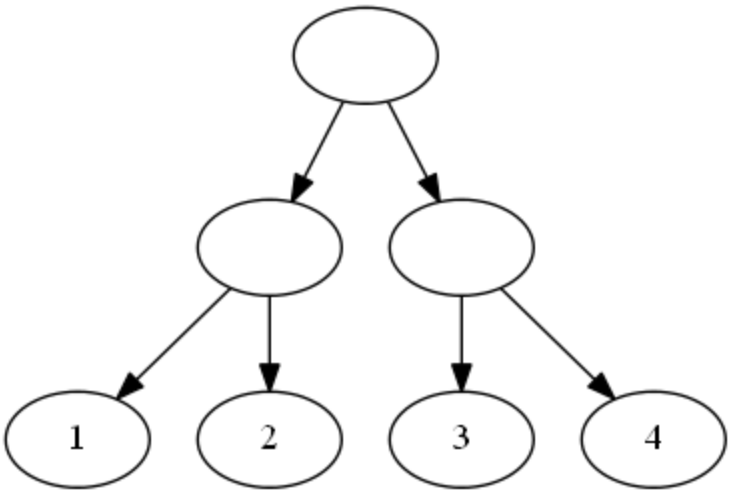

Figure 3.2

… can be represented as:

((1, 2), (3, 4));

Node labels may be in one of two forms: quoted and unquoted. Unquoted node labels are composed of one or more UTF-16 characters *except*:

the characters: ( ) [ ] : ; # ‘ , .
space
tab (\t)
carriage return (\r)
newline (\n)

Further, the underscore character “_” within an unquoted node label is interpreted to be a space. As such the Rich Newick string:

(red_node, black_node)green_root;

contains three labels “red node”, “black node” and “green root”.

Quoted node labels always begin and end with the single quote character “‘“ and contain zero or more UTF-16 characters excluding the characters:

quote (“‘“)
carriage return (\r)
newline (\n)

except for the case where two quote characters appear adjacently (excluding the first and last quote characters as one of the adjacent characters). Two adjacent quote characters are interpreted as a single quote character literal within the body of a quoted label and not as the termination of the quoted label. For example, to create a three node phylogenetic X-network where the root contained the label:

The dog’s tail wags.

and with X = {1, 2} we would create the string:

(1, 2)‘The dog’’s tail wags.’;

Note how in the quoted label case spaces can be included in the node label without having to use the underscore character. In a quoted label the underscore character is interpreted literally as “_” in the body of a label.

4. Edge Labels

Within a phylogenetic network we may wish to associate certain attributes with a given edge--namely branch length, support, and probability values. Branch length values may be any decimal number while support and probability values may be only any decimal number between 0 and 1 inclusive.

Consider the phylogenetic X-network:

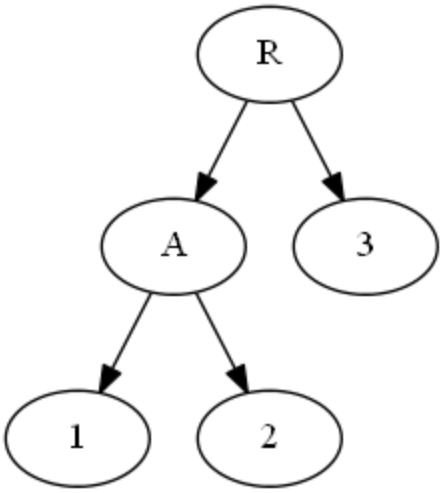

Figure 4.1

To associate values with a given edge we first locate the position in the Rich Newick string that corresponds to the node incident to the given edge within the node list that defines that edge. For example, to associate a branch length with the edge connecting *A* and *2* in the example above we would locate the position of *2* in the Rich Newick string since *2* (and not *A)* appears in the node list that defines the edge connecting *2* and *A*. (*A* is also a member of another node list--namely the list connecting *R* with *A*--but since this node list does not define the edge connecting *2* and *A* the node *A* is not the proper choice for our example.) Next we append to the node's label a colon followed by the desired decimal branch length--for example, 30.8:

((1, 2:30.8)A, 3)R;

… which could be diagrammed as the rooted tree:

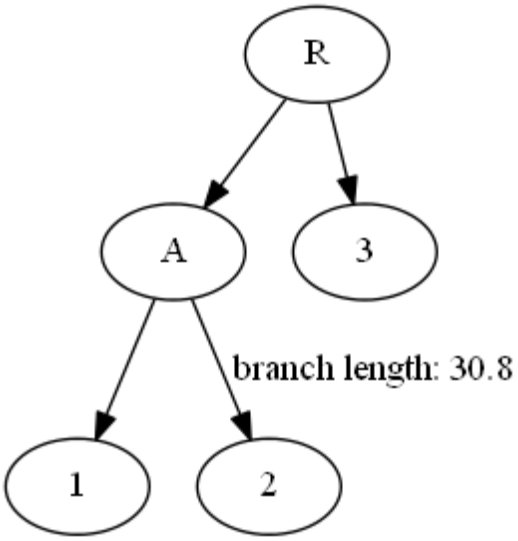

Figure 4.2

(The placement of edge attributes for a hybrid node is slightly different. See the section on hybrid nodes for discussion and examples.) In the event that the node in question is unlabeled we place the colon and value where a node label would normally appear. So, for example, if in a similar network to the example above the node corresponding to *A* was unlabeled we could append the branch length 7 to the edge from *R* to that corresponding node with:

((1, 2:30.8):7, 3)R;

Support values and probability values can follow in similar form by appending additional colons and values to branch length values. For example, a unrooted phylogenetic X-tree with X = { 7, 9 } and a single edge consisting of branch length 500, support .8 and probability 1 would be represented as:

[&U](7:500:.8:1, 9);

where branch length appears first, support second, and probability third.

Any combination of branch length, support and probability may be attributed to an edge using the following syntax for example node *E*:

branch length only: E:*length*

branch length, support:	E:*length*:*support*
branch length, support, probability:	E:*length*:*support*:*probability*
branch length, probability:	E:*length*::*probability*
support only:	E::*support*
support, probability:	E::*support*:*probability*
probability only :	E:::*probability*

It should be noted that probability values other than 1 are only permitted for networks of the phylogenetic X-network form and are only attributed to in-edges incident to a hybrid node (as discussed later).

5. Representing Hybrid Nodes

Hybrid nodes are those nodes in a phylogenetic X-network with in-degree >= 2. Consider the following phylogenetic X-network with hybrid node *Z*:

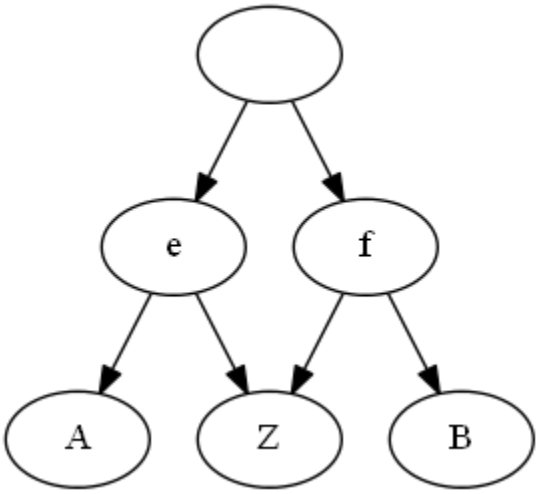

Figure 5.1

... which could be encoded by the Rich Newick string:

((Z#1, A)e, (Z#1, B)f);

A hybrid node is identified in a Rich Newick string by use of the “#” character followed by a a positive integer called the *hybrid node index*. The hybrid node index may

optionally be prepended by a single *hybrid node type* which in turn may be the characters “R” (to denote recombination), “H” (to denote hybridization) or “LGT” (to denote lateral gene transfer). For example:

((Z#H1, A)e, (Z#H1, B)f)

Because *Z* is a direct successor of both *e* and *f* it appears in the node lists of *both e* and *f*. When creating a hybrid node the node label, hybrid type and hybrid node index must match exactly for each occurrence of the node in all node lists. However, successors of the hybrid node may only be included in the node list of a single occurrence of the hybrid node. For example, to encode the following phylogenetic X-network:

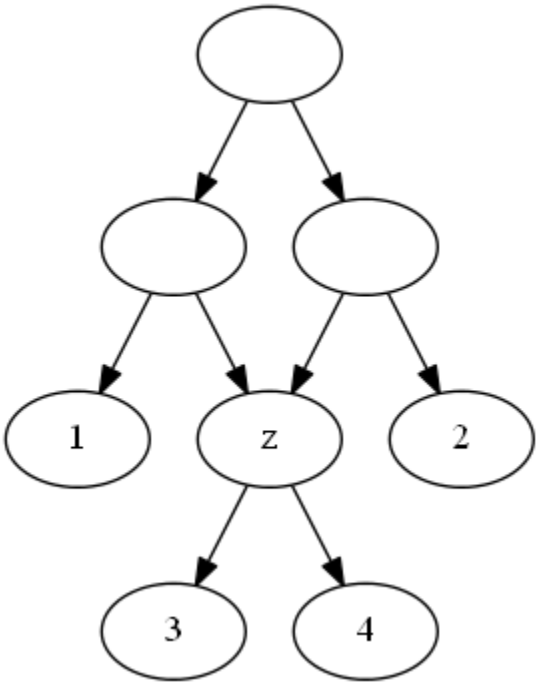

Figure 5.2

...we could use the Rich Newick string:

(((3, 4)Z#H1, 1), (Z#H1, 2));

or

((Z#H1, 1), ((3, 4)Z#H1, 2));

but not

(((3)Z#H1, 1), ((4)Z#H1, 2));

Edge attributes for hybrid nodes take the same form as for non-hybrid nodes except that they are append to the hybrid node index instead of a label (or node position). For example:

((Z#H1:200:.8:.3), (Z#H1:100:.9:.7));

Note how the edge attributes appended to the hybrid node index can differ for each syntactical representation of the hybrid node. This is on account of the fact that these edge attributes apply to two different edges in the network--namely the edge connecting *A* and *Z* and the edge connecting *B* and *Z*. It should also be noted that if a probability attribute is specified for one in-edge of a hybrid node it is considered an error for any other in-edge of the hybrid node to omit a probability attribute. Further, the sum of all probability attributes of all in-edges of a hybrid node must be 1. If all in-edges of a hybrid node do not specify any probability attributes then the probability value for each of the *n* in-edges is assumed to be 1 / *n*.

6. Comments

Comments may be embedded within Rich Newick strings by use of matching “[“ “]” characters with the exception of the sequence “[&R]”, “[&r]”, “[&U]” or “[&u]” appearing at the beginning of the string. For example, the following Rich Newick string utilizes comments:

[this is a comment](A, B [and another comment])R;

Comments in Rich Newick Strings may be nested and/or span lines.

7. Context Free Grammar

It is necessary but not sufficient for valid Rich Newick strings to conform to the following EBNF grammar:

```
network =                   ROOTAGE_QUALIFIER? descendant_list? network_info ';';

descendant_list =           '(' subtree (',' subtree )* ')';

subtree      =              descendant_list network_info
             |              network_info;

network_info=               node_label? hybrid_node_qualifier? branch_length?
                 |          node_label? hybrid_node_qualifier? branch_length      bootstrap
                 |          node_label? hybrid_node_qualifier? branch_length      bootstrap   probability
                 |          node_label? hybrid_node_qualifier? branch_length      ':'         probability
                 |          node_label? hybrid_node_qualifier? ':'                bootstrap   probability?
                 |          node_label? hybrid_node_qualifier? ':'                ':'         probability;
```

```
branch_length=                  edge_label;

bootstrap=                      edge_label;

probability=                    edge_label;

node_label=                     text;

hybrid_node_qualifier=          '#' UNQUOTED_ALPHA_TEXT? DECIMAL_NUMBER;

edge_label=                     ':' DECIMAL_NUMBER;


text=                                   QUOTED_TEXT
    |                                   UNQUOTED_ALPHA_TEXT
    |                                   DECIMAL_NUMBER;

ROOTAGE_QUALIFIER=                      '[' '&' ('r'|'R'|'u'|'U') ']';

DECIMAL_NUMBER=             ('0'..'9')* '.'  ('0'..'9')*
                        |   ('0'..'9')+ ('.' ('0'..'9')*)?


UNQUOTED_ALPHA_TEXT=    -('(' | ')' | '[' | ']' | ':' | ';' | '#' | '\'' | ',' | ' ' | '.' | ('0'..'9')  | '\t' | '\r' | '\n')*;

QUOTED_TEXT=                            '\'' (-('\n' | '\r' | '\'') | ('\'\'') )*  '\'';

NESTED_ML_COMMENT=                      '[' (-('[' | ']') | NESTED_ML_COMMENT)* ']'

WHITE_SPACE  =          ( ' ' | '\t' | '\r' | '\n');
```

8. Context Sensitive Rules

Valid Rich Newick strings must conform to the following context sensitive rules:

1. Support values must be between 0 and 1 inclusive.
2. Probability values must be between 0 and 1 inclusive.
3. All leaf nodes of a phylogenetic network (i.e. phylogenetic X-tree or phylogenetic X-network) must have a node label.
4. For phylogenetic X-networks, if any in-edge of a node contains a probability value then all in-edges of that network node must contain a probability value.
5. For phylogenetic X-networks, if any in-edge of a node contains a probability value then the sum of all probability values for all in-edges of that node must be 1.

6. For phylogenetic X-trees, if any edge contains a probability value then the value must be 1.

7. For a phylogenetic x-tree, if the outermost node list contains exactly two nodes (irrespective of the number of nodes in those node's subtrees) then no node label or edge attributes may appear at the network_info position appending that node list.

8. Each syntactical occurrence of a hybrid node (as identified by its index) must match with respect to node label (or lack thereof) and hybrid node type (or lack thereof).

9. Each hybrid node must appear in 2 or more node lists.

10. Only one syntactical occurrence of a hybrid node may append a node list.

9. References

1. Cardona, G., Rosselló, F., & Valiente, G. (2008). Extended Newick: it is time for a standard representation of phylogenetic networks. *BMC Bioinformatics*, *9*(1), 532. BioMed Central. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/19077301

2. Semple, C. and Steel, M. Phylogenetics. New York: Oxford, 2003