Quick Path Finding – Quick Algorithmic Solution for Unambiguous Labeling of Phylogenetic Tree Nodes

Piotr Płoński[1,2], and Jan P. Radomski[1*]

[1] *Interdisciplinary Center for Mathematical and Computational Modeling, Warsaw University, Pawińskiego 5A, Bldg. D, PL–02106 Warsaw, Poland*
[2] *Institute of Radioelectronics, Warsaw University of Technology, Nowowiejska 15/19, PL-00-665 Warsaw, Poland*

Keywords: "quick path finding algorithm", "neighbor joining algorithm", "phylogenetic analysis", "phylogenetic tree", "Newick format", "automatic node labeling"

Abstract

Ever increasing amounts of genetic information stored in sequential databases require efficient methods to automatically reveal their phylogenetic relationships. A framework for in silico *unambiguous analysis of phylogenetic trees, based on information contained in tree's topology, together with its branches length, is proposed. The resulting, translated tree has all nodes labeled, with no constraints on nodes' degree, and the subsequent finding of evolutionary pathways from the QPF-translated tree is robust and straightforward. Main features of the method are: small demands on computational time, and the ability to analyze phylogenies obtained prior to the proposed QPF analysis by any traditional tree-building technique.*

1. Introduction

The extremely large comparative genomic data sets available today pose escalating computational challenges for their automatic phylogenetic analysis. Existing methods can be divided in four groups**:** maximum parsimony (Felenstein, 1978), maximum likelihood (Felenstein, 1981), distance-based (Saitou and Nei, 1987), and Bayesian based (Larget and Simon, 1999), however, all the above are only heuristics as constructing proper phylogenetic tree is a NP-hard problem.

A phylogenetic tree is a way to visualize sequences' evolutionary relationships, and usually it contains information about sequences used to built tree's topology, together with branches length. From mathematical viewpoint, a phylogenetic tree is a graph, which contains no cycles (Deo, 1974). This means that between every node in a graph there exists one and only one path. Vertices of a tree represent sequences. Edges connecting vertices, describe events between sequences. Edges can be weighted or unweighted; which means they can express events quantitatively or not. Number of adjacent edges in node is a node's degree. Nodes of degree one are leaves, nodes with degree two or greater are internal nodes. Graph can be directed – with a root, or undirected – without a root. Directed graph shows the evolution time.

However, building a phylogenetic tree is only a beginning of a process to analyze

* E-mail addresses: pplonski@icm.edu.pl, and janr@icm.edu.pl *corresponding author*

sequences' set. Many times we seek more**:** molecular rates, population parameters, or even evolutionary pathways – to obtain that information, inference from molecular phylogenies could be used. Some methods require only tree topology, for instance (Slowinski and Guyer, 1989) to compare diversification rates of sister clades, or to measure tree's balance (Kirkpatrick and Slatkinm, 1993). On the other hand, there are methods that utilize only branches' length mainly for testing birth-death coalescent processes (Nee *et al*., 1994), to estimate number of missing taxa rates of population growth (Pybus *et al*., 2002), or to estimate rates of molecular evolutions (Rambaut, 2000). Finding evolutionary pathways seems to be much more challenging problem, because there are virtually no methods of finding evolutionary pathways by exploiting only information from traditionally made phylogenetic trees. Some new methods were proposed for evolutionary pathway studies**:** (Hasegawa *et al*., 2009) presents vSPA method, which trace evolution of serial-sampled sequences divided in clusters; (Ren *et al*., 2003) developed method that used Neighbor Joining distance matrix to construct longitudinal phylogenetic trees.

Inference of evolutionary pathways from traditionally generated phylogenetic trees is a difficult task for two reasons: first, in a traditional phylogenetic tree all sequences are shown in a tree as leaves – the internal nodes are not attributed to any particular sequences. We can interpret such a tree as showing only sequences that had no descendants. However, it is more complicated, because even if we analyze a set with sequences, which have their descendants in this set, they will be shown as leaves. Thus, construction of traditionally generated phylogenetic tree does not agree with such a tree interpretation, since internal nodes are treated as ancestors of leaves (Baum, 2008). Secondly, such traditional tree is often a binary tree, which means that nodes' degree is three or less. It is significant constraint as an ancestor can have any number of immediate descendants. Lets consider a sequence that has more than two descendants, all of them will be shown on tree as an assemble of auxiliary internal nodes and leaves. Binary character of such a tree is a consequence of clustering algorithms, which merge sequences into pairs. Keeping all these drawbacks in mind, it means that finding evolutionary pathways will grow to a problem of finding all ancestor-descendant pairs in a tree, which would require to always check all the nodes for every sequence – such an approach will be computationally rather demanding.

The aim of this work was to present an alternative algorithm, which we would like to call the Quick Pathway Finding (QPF), for constructing a graph with all nodes labeled, and with no node degree constraints – after generating a QPF translated three the subsequent task of inferring evolutionary pathways would be a computationally simple and rather quick endeavor, which moreover can be performed in a fully automatic manner.

2. Interpretation of Nodes

2.1 Leaves interpretation

Nodes in a tree can be divided in two types: [a] internal nodes, and [b] leaves. Internal nodes are sequences, which are distinct but have at least one offspring. Nodes with no offspring are presented as leaves. In particular, an offspring can be an assemblage

of internal nodes and/or leaves. Lets consider only directed trees, with branches lengths between sequences equal to a genetic distance. Different possible characters of leaves in a tree are shown in **Figure 1** – for clarity presented both as cladograms, to show construction of a tree, and phylograms to show distances between sequences. All trees in **Fig. 1** arise from only two sequences, and can be encoded in the Newick format by the same rule ($A:d_A$, $B:d_B$) – the only differences are in their edge weights, which is why they all have the same cladogram, but different phylograms. In all cases it can be concluded that A and B are closely related, and have a common ancestor. In the first tree, d_A is zero and d_B is equal to distance between A and B. Note that sequence A and B must have the same ancestor, from which follows that A must be a parent of B. Only in this solution the distance between A and its parent is zero, and distance between B and its parent is equal to distance between A and B. In the second case, d_A and d_B are equal or greater than one unit distance (distance between sequences which differ by only one mutation). They have common ancestor, which is not present in the analyzed set. All of them have an offspring. In the third tree, d_A and d_B are close to zero, this means that each sequence should be an ancestor, and a descendant at the same time, which indicates that sequence A and B must be the same. Therefore, just by using the knowledge about branches length the family character of all nodes in a tree can be inferred.

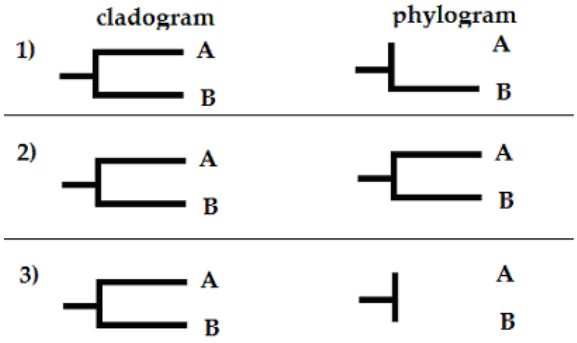


Figure 1. Different types of nodes in a phylogenetic tree.

2.2 Character of Internal Nodes

Unfortunately, in traditionally build phylogenetic trees all sequences are treated as leaves, no matter if they have any offspring or not. Such trees treat internal nodes as auxiliary nodes to hold other nodes, or as information about potential ancestor sequence, which is not present in the analyzed set (**Figure 2a**). This could pose a potential trap in designing algorithm analyzing evolutionary pathways, as one would expect ancestors to be closer to root than descendants (in the number of the edges between them). Only sequences on terminal branches (leaves), and the root are observed. All internal branches of the tree are not observed – if necessary, such sequences have to be estimated by the reconstruction (Ren *et al.*, 2003). Therefore, a possible other role of internal nodes in traditionally build trees might be to represent potential ancestor sequence[s]. This situation can happen only when all adjacent edges' weights are greater than an unit

distance – a necessary condition, as in any other edge weights' configuration such an internal node will be just an ordinary auxiliary node. The ambiguity of leaves' character, and internal nodes' meaning in traditional phylogenetic trees contributes to difficulties in efficient analysis of evolutionary pathways *in sillico*. To alleviate this problem was the goal of the present study. The next section describes an algorithmic recipe, which considers character of all nodes in a traditional tree, and then attributes them with an appropriate role in a translated evolutionary tree (**Figure 2b**).

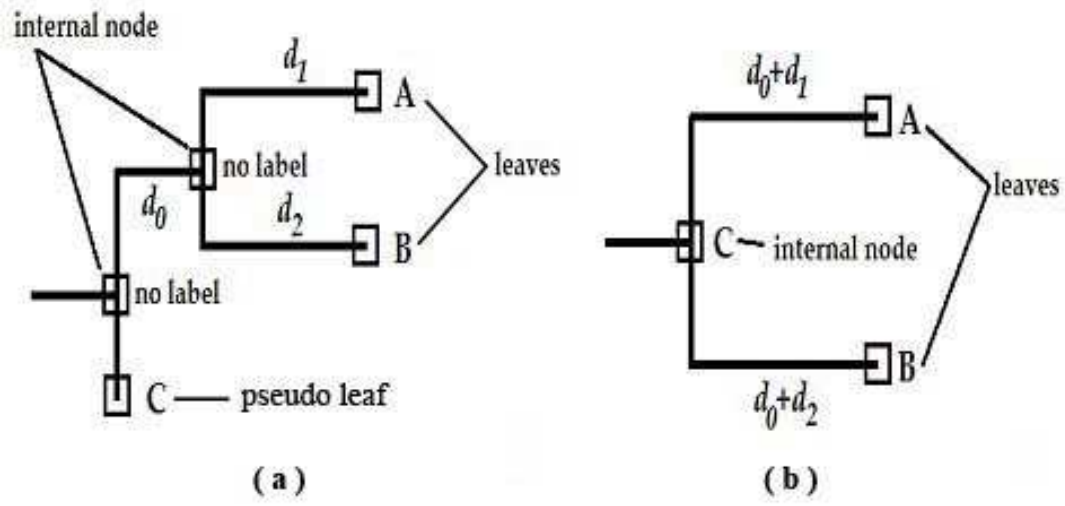


Figure 2. A schematic phylogenetic tree; (**a**) typical reconstructed phylogenetic tree, which contains internal nodes without labels – written in Newick notationa as: ((A:d_1, B:d_2):d_0, C:0); (**b**) phylogenetic tree with all nodes labeled obtained from the tree in (**a**), written in Newick notation as: ((A:d0+d1, B:d0+d2)C:0); in both panels **C** is the ancestor of **A** and **B**.

3. The Algorithm

The proposed algorithm consists of two stages**:** first, to rebuild traditional phylogenetic tree, and then to find evolutionary pathways in the resulting QPF tree. At the beginning algorithm reads the input phylogenetic tree *T=G(V, E)* written in the Newick format. The *BFS* algorithm is used to find distance R_i between the root and all other nodes, expressed as a number of edges. In the next step, nodes are ordered by descending values of R_i. Translation to a new tree *T'=G(V', E')* begins from the largest value of R_i. For each translated node, two types of behavior can be distinguished, depending on node's degree. For each node a family character of the leading edge weight have to be decided, as described in section 2.1. As a threshold we assume half of unit distance (*c.f.* the discussion in section 5. for details). When node's leading edge weight is smaller than *h* we treat it as a node without offspring (a child); for weight equal or greater than *h* we treat it as a node with offspring (a parent). Then a new node is made in the QPF tree *T'*. It has the same label and leading edge weight as the original node, and its corresponding node in QPF needs to be remembered – this will be required in further algorithm's steps. All internal nodes convey the information about tree topology, and during internal node's translation we infer newly created nodes' position. Before that, it

is necessary to designate for all the nodes, held by the considered node, their family character using procedure described above. In a case when an internal node holds nodes, that after translation would acquire the following *scheme***:** one of them is a parental node (marked as *A*), and the rest are child-character nodes, for all child nodes we need to remember as their ancestor this parental node *A*. After that, the *A* node's edge weight is updated by adding the internal node's leading edge weight, and we need to remember for that internal node its corresponding node as *A*. Such a treatment is necessary, because the parent node should be present also when all further translations will be considered. If an internal node holds only nodes, which after translation would act as child nodes, then all of them are moved to the internal node, which already holds the node under consideration. Before moving nodes we increase all nodes' leading edge weights by the leading edge weight of the considered internal node. It should be pointed out, that it is not possible to contain in one internal node two nodes with a parent character, as we only consider unique sequences in the analyzed set. After translation of all nodes of the input tree we end with pairs of ancestor-descendant sequences, from which it is relatively easy to reconstruct evolutionary pathways. The resulting translated tree is then saved in the Newick format.

3.1 Algorithm's steps

1. Read the tree *T=G(V, E)*, remember for every node *v* in *V*, the nodes which it holds marked as *C*, and node which holds *v*, marked as *P*, and leading edge weight, marked as *D*;
2. Use the BFS to compute the number of edges R_i between root and all nodes *v* in *T*;
3. Order nodes *v* by descending R_i, and push them to priority queue *L*;
4. Repeat steps 5, 6, 7 until *L* is not empty;
5. Get node *v* with largest R_i from *L*, and pop it back from *L*;
6. If *v* has a label, make a node *q* in *QPF* tree with label, and leading edge weight as *v*, remember for *v* corresponding node *q* in *QPF*;
7. If *v* has no label, decide family character for every node in *C* like in the step 8:
    a) if among the nodes *C* exists one with a parent character marked as *A*, then we need to remember for all the nodes in *C* their ancestor as *A*; and then to add for the *A* the leading edge's weight – that is the weight of *v;* and finally to remember the *A* as a corresponding ancestral node of *v*;
    b) if in the nodes *C* all are children, we move them to the node *P*; then the children, for all moved nodes, need to add to their leading edges' weight the distance of considered node's leading edge weight *D*;
8. Decision step for node *v*:
    a) if *v* incoming edge's weight < *h*, then node's *v* character is a parent;
    b) if v incoming edge's weight >= *h*, then node's *v* character is a child;

For the node analysis we make use of geometrical algorithm known as a sweep line (Corment et al., 1989). First, all nodes are projected to one dimension - R_i, and then we start sweeping (translate) from the highest R_i, because nodes (which such internal node holds) will always have larger R_i. From this follows that during translation of such

internal node its children were already translated. The diagram of a tree sweeping in shown in **Fig. 3**, and the pseudocode of QPF algorithm is shown in **Fig. 4**.

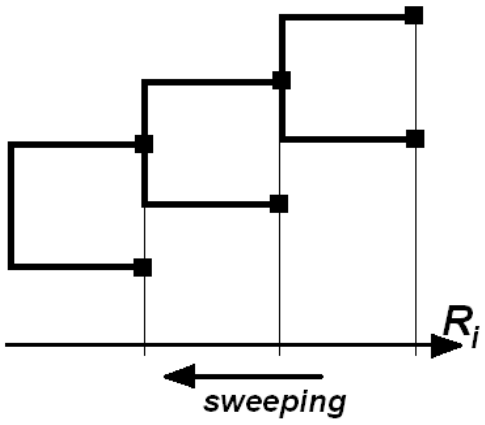


Figure 3. The diagram of tree sweeping.

For finding evolutionary pathways present in the resulting QPF tree, between each node and the root DFS-like recursion is used. Due to necessity of sorting nodes in one of the steps, the complexity of the algorithm is O(V log V), where V is number of nodes in the input, traditional phylogenetic tree. The obtained complexity is much better than if we would search for a pathway between every node and its parent. For this purpose we could use Dijkstra algorithm, with complexity for rare graphs O(E log V), and then for all nodes the complexity will became O(V*E log V).

```
struct Node {

label;  // label if exists
p;      // parent
c;      // array of holding nodes
d;      // leading edge's weight
ri;     // number of edges from root
tr;     // pointer to the corresponding translated node in QPF
ch;     // node's character in QPF (a parent or a child)

}

Input:
      Tree T(v)                                    // v - nodes of a tree
      Treshold h                                   // half of unit distance

Algorithm:
      Build an empty tree QPF;
      DFS(T); // start from root, calculate v->ri
      Push all v to priority queue L, order by v->ri, descending;

      Until L is not empty:
            Get v with the highest v->ri;
            If v->label exists:
                  Make an empty node q;
                  q->label = v->label;
                  q->d = v->d;
                  v->tr = q;
                  Add q to QPF;
            IF v->label does not exist:
                  For each v->c:                   // nodes translation
                        If v->c[i]->tr->d >= h:
                              v->c[i]->tr->ch = child;
                        If v->c[i]->tr->d < h:
                              v->c[i]->tr->ch = parent;
                  If exists v->c[i]->tr->ch equal to parent, mark it as A:
                        For each v->c, with v->c[i]->tr->ch equal to child:
                              v->c[i]->tr->p = A;
                              A->c.Add(v->c[i]->tr);
                        A->d += v->d;
                        v->tr = A;
                  If all v->c[i]->tr->ch equal to child:
                        For each v->c:
                              v->c[i]->tr->d += v->d;
                              v->p->c.Add(v->c[i]);
            Pop v from L;

      BFS(QPF); // start from leaves (q->ch equal to child), find pathways

Output:
      the translated QPF tree                      // in Newick format
      the description of pathways                  // in a text file
```

Figure 4. The pseudocode of the QPF algorithm. The data structure is described first – the same structure is used by the algorithm for both traditional trees and QPF trees. The operator "->" denotes that a structure's variable is called. The operator "**.**" means that a function described after the „dot" is called to operate on a given variable.

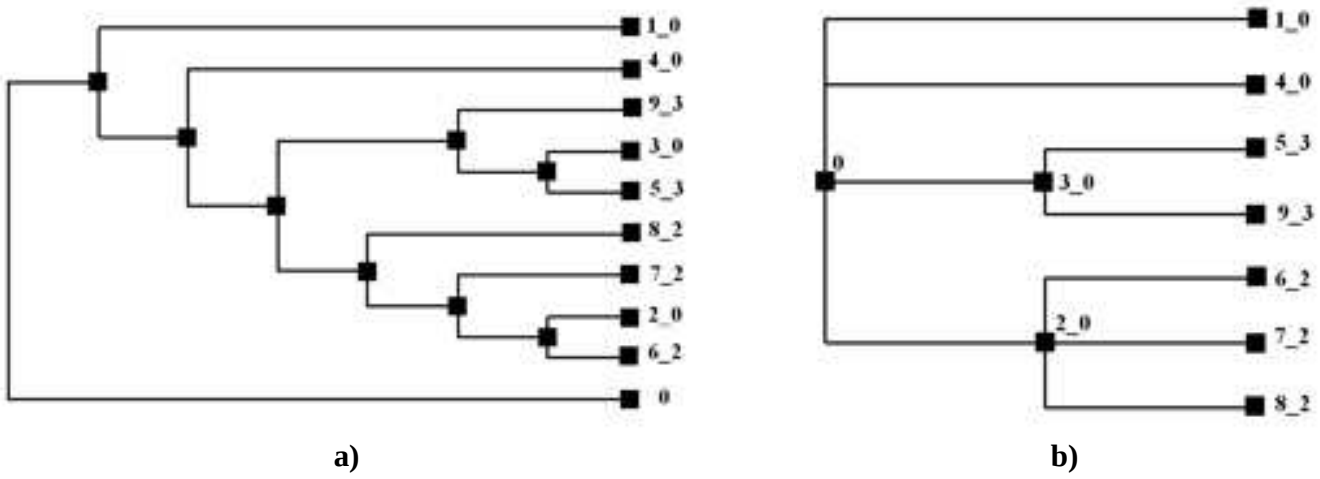
4. Performance Results

To examine the performance of the algorithm several sets of artificial sequences were generated. All sequences were represented by strings of binary characters – for the

root sequence all characters were set to “0”. Progeny sequences were then obtained by random changes from ‘0’ to ‘1’, resulting in perfect phylogeny (Gusfield, 1991), as only one change at each position was applied. The number of descendants for each sequence was determined randomly. For a set of sequences thus created their true evolutionary pathways are of course always known. Nine sets were generated, with sizes N = {10, 20, 50, 100, 200, 500, 1000, 2000, 5000} sequences respectively. For each set, its phylogenetic tree was computed by the Neighbor-Joining algorithm (Saitou and Nei, 1987), and re-rooted. The NJ trees were then translated by QPF, and their evolutionary paths were calculated as described earlier. **Figure 5** shows a traditional phylogenetic tree, and a corresponding translated QPF tree for a set of ten sequences. In the **Table 1** performance of the algorithm is shown, confirming our estimate of the procedure’s complexity, which is very nearly linear.

No. of sequences in the set	Time [seconds]
10	0.022485
20	0.016908
50	0.018563
100	0.037877
200	0.061501
500	0.132665
1000	0.252395
2000	0.547458
5000	1.246603

Table.1. The number of sequences in a set, and the corresponding execution times.



a) **b)**

Figure 5. Panel (**a**) – the cladogram of a phylogenetic tree obtained by the Neighbor Joining method; panel (**b**) **–** the cladogram of the QPF tree obtained from the tree shown in the panel (**a**). Both trees were made from a set of 10 sequences. Each label consist of the sequence's number, and after underscore '_', the number of the parent's sequence, root sequence has the label "0".

5. Robustness

5.1. Data and methods

To examine an impact of tree imperfections on the evolutionary pathways obtained by the QPF algorithm, two kinds of artificial sequence sets were generated using Monte Carlo methods. In the first type of sets, all sequences were binary, coded, with chain lengths of 2000 characters each (this corresponds roughly to 1000 nucleotides). Between an ancestor and each descendant there was always exactly one mutation, and the number of descendants was randomly drawn from the *(0; D)* range, where *D* denotes maximum number of descendants. There was an additional condition on a number of children drawn – to always produce a predefined number of sequences [when a number of generated sequences was smaller than requested, then as the last generated sequence we have drawn randomly the number of children from *(1; D)*]. All mutations were randomly distributed. Mutational event comprised of changing a randomly selected position to an opposite character. Resulting sequences were requested to be unique.

The second kind of sets comprised of pseudo-real nucleotide sequences (**L**=1701 nucleotides each), the original seeding sequence was taken from GenBank, and then descendants were Monte Carlo generated from it, in order to have a full information recorded concerning all evolutionary pathways present in each set. Mutation events were allowed to occur if a random number $\boldsymbol{p_1}$, drawn from an interval (0;1) was smaller than $1-exp(m * L)$, where ***m*** is mutation frequency coefficient. In every mutation step, number of descendants of a sequence was examined, and when random number $\boldsymbol{p_2}$, drawn from (0, 1) was smaller than $exp(r * C)$, then a new sequence was created. The parameters: ***C*** is

the number of children that a sequence already has, and ***r*** is coefficient which regulates the number of descendants. All mutations were equally probable, and uniqueness of sequences was not enforced.

Four collections (**A**-**D**) of sequence sets were generated by Monte Carlo routines as described above, for set sizes *N={10, 20, 50, 100, 200, 500, 1000}***:**

A. 500 binary sequences sets for each size in *N*, with about 50% leaves to internal nodes ratio in their trees, *D=3*;

B. 100 binary sequences sets for each size in *N*, with about 30% leaves to internal nodes ratio in their trees, *D=5*;

C. 100 binary sequences sets for each size in N, with about 60% leaves to internal nodes ratio in their trees, *D=2*;

D. 100 pseudo-real nucleotide sequences sets, with about 50% leaves to internal nodes ratio in their trees; parameters: *m=0.001, r=0.15.*

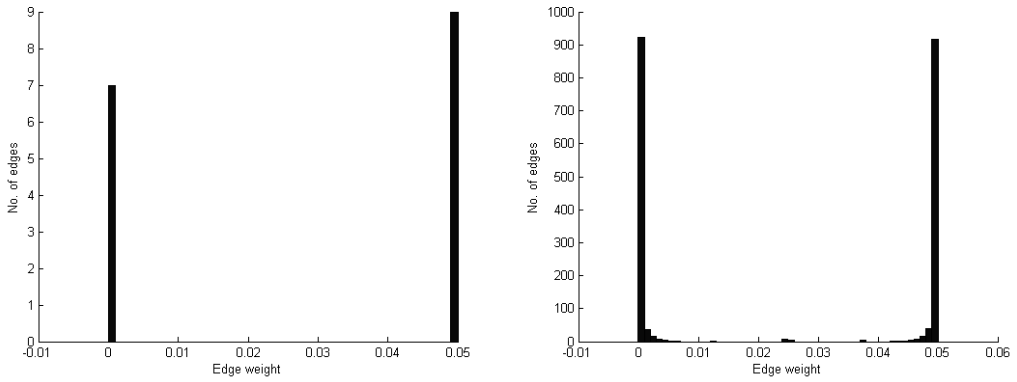
5.2 Trees quality

The efficiency of evolutionary pathways' finding depends mostly on the input phylogenetic tree's quality, and for large sequence sets might still pose a problem. The question arises as to what extent the QPF algorithm can cope with traditionally built trees' imperfections. In the QPF approach edge's weights distribution plays a crucial role as it is used to infer family character of nodes. In a perfect phylogeny's tree all edge weights should be multiples of a unitary distance, so the pathway length between the nodes strictly represents distance between them. In a perfect phylogeny there are no two identical mutations at the same position. However, when analyzing a real sequence set such situation is not uncommon – the same mutations can occur at the same positions. Therefore, the edges' weights distribution will be disturbed, and not all weights would be multiples of unitary distance. Such disturbed distribution depends on few factors. A bigger chance for a noisy tree occurs when: (i) the analyzed set consist a large number of sequences; (ii) sequences chain's lengths are small; (iii) the number of characters which code each position in the chain is small; (iv) the mutations tend to be not distributed randomly throughout the chain (and also throughout a time domain – mostly due to biases in the time of samples isolation, and the numbers of isolated sequences).

5.3 QPF robustness

The typical edge weights' distributions for sets of 10 sequences, and 1000 sequences are shown in **Fig. 6a** and **6b** respectively. For the former set, a chance to have a noisy tree is rather small, so a bimodal distribution with the two clear peaks is observed (at values which are multiplies of unitary distance). On the other hand, as the number of sequences increases, the chance to have a noisy tree grows. The **Fig. 6b** shows, that for

the set of 1000 sequences, there were some with their values slightly different from multiplies of unitary distance. The improper distribution would have, of course, an impact on a tree topology, and quite often leads subsequently to a misleading tree's construction. Two peaks are observed because, in generated sets, all sequences differ always by exactly one mutation. In **Fig. 6a** nine edges are observed with their weights equal to unit distance, each edge represents mutational event between sequences (to generate 10 sequences there must be 9 mutations); additionally there are seven edges with zero weights – they represent auxiliary nodes, or nodes of a parent character. When the number of mutations grows, the number of observed peaks increases respectively. In contrast, the distribution of edges' weights for the QPF tree is significantly different, as in this case only edges' weights are observed, representing mutations between sequences directly. So, should there were only single mutations, then the QPF edges' weights distribution must have only one peak at an unit distance, there are no any edges' weights equal to zero in QPF, as auxiliary nodes do not appear in the QPF tree.



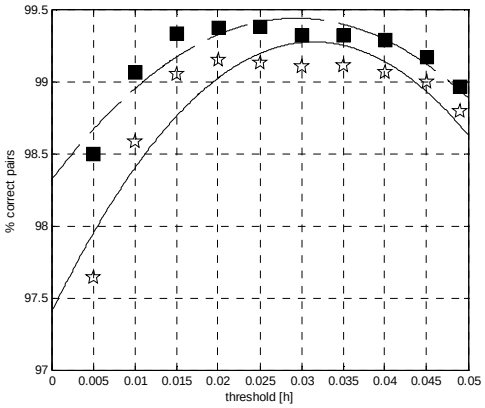
a) **b)**

Figure 6. The distributions of the edge weights in traditional NJ phylogenetic trees from the set of: (**a**) 10 sequences; and (**b**) 1000 sequences.

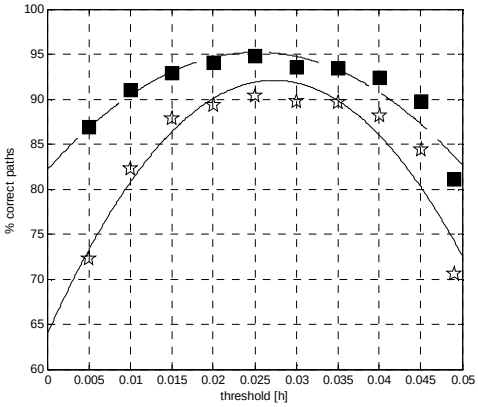
In a case of improper choice of the threshold value, it could lead to a singular situation – there would be (in the QPF topology-decision step) two nodes with parent's character, but for a descendant only one parent is possible. This can happen when setting a threshold value very close to a unit distance, or a set of sequences is not unique. Both situations would appear abnormal to the QPF algorithm. Should this happen nonetheless, then an error message is generated, to the effect that a decision for further analysis of edges' weights is necessary. In a case when leading edges' weights of two nodes considered are equal, this would mean that their sequences are identical, and consequently one sequence should be excluded from any further analysis. On the other hand, when leading edges' weights are different, algorithm makes a decision, that one of considered nodes is a parent (the one with a lower edge weight), and the other node is treated as a child.

Therefore, unless a traditional phylogenetic tree is excessively noisy (in as sense of both: edge's weights distribution, and tree's topology) it is not a problem to obtain an

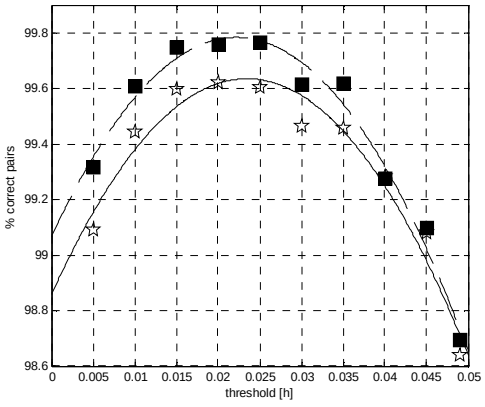
acceptable efficiency of reconstructed evolutionary pathways by using a judiciously adjusted threshold decision value of ***h.*** As the accuracy of the QPF algorithm depends on the value ***h***, to check the actual QPF performance, the correct ('true') evolutionary pathways were recorded for every generated artificial sequence set, their phylogenetic trees were obtained by the Neighbor Joining method (Saitou and Nei, 1987), each tree was then re-rooted from it's primary ancestor, and finally the QPF translation was performed several times using different threshold ***h*** values, to compare the 'true' evolutionary pathways with the QPF-reconstructed ones. The unit distance is 1/2000 * 100 = 0.05 for binary sets. It is not necessary to check values of ***h*** equal or near 0 and 0.05 for binary sets, because for such thresholds, all nodes would be treated either like a child, or like a parent. Accordingly, values of ***h*** = *{0.005; 0.01; 0.015; 0.015; 0.02; 0.025; 0.03; 0.035; 0.04; 0.045; 0.049}* were examined.



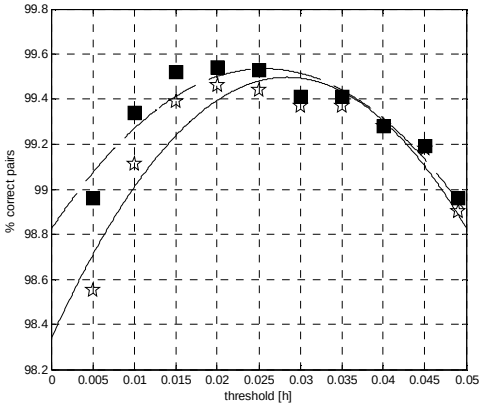
(**a**)



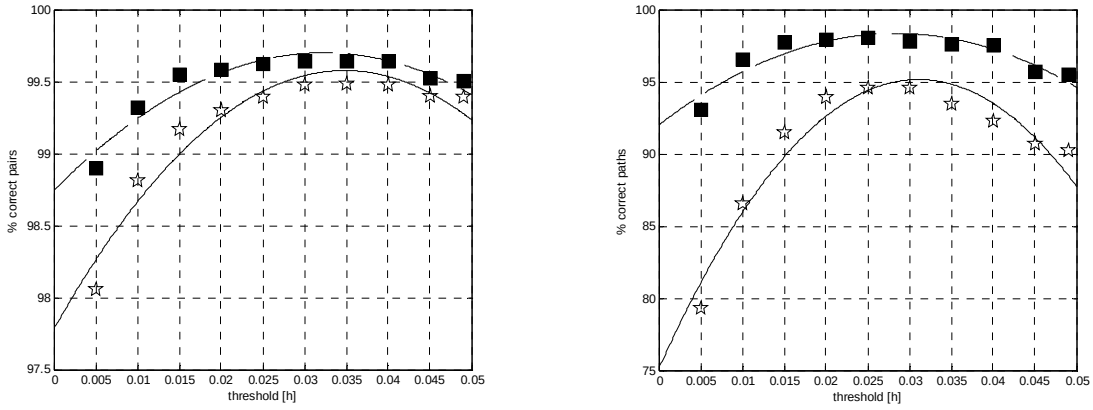
(**b**)



(**c**)



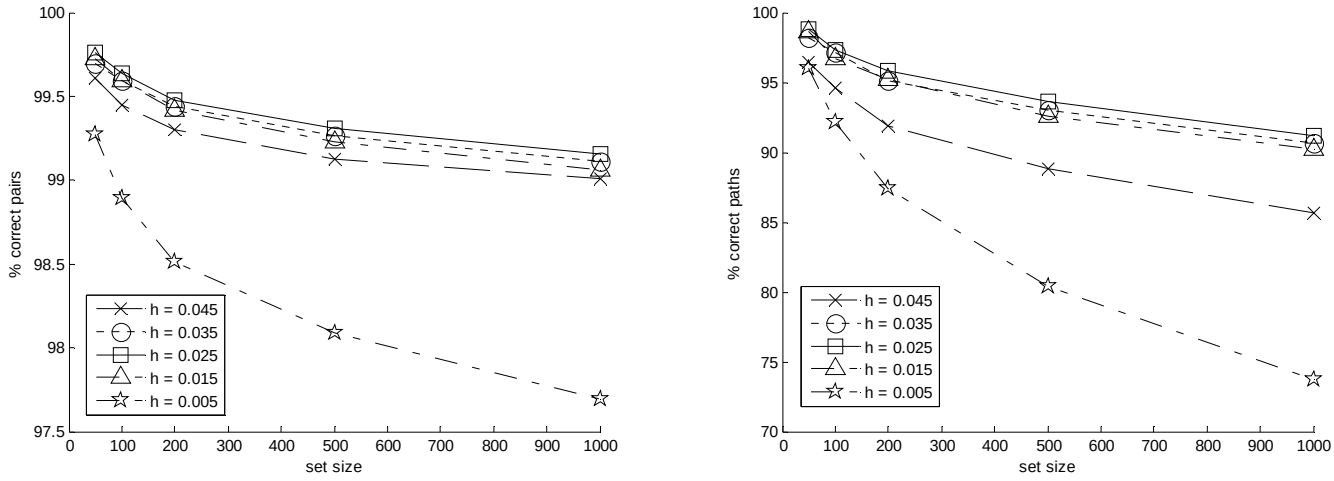
(**d**)



(**e**) (**f**)

Figure 7. The percentages of correctly reconstructed ancestor-descendant pairs, and evolutionary pathways as a function of the QPF threshold ***h*** *= {0.005; 0.01; 0.015; 0.015; 0.02; 0.025; 0.03; 0.035; 0.04; 0.045; 0.049}*, for set sizes *{200, 1000}*. Four collections (**A**-**D**) of sequence sets were used (as described in text, *vide supra*), generated with different ratios between internal and terminal nodes of their phylogenetic trees. For the collection **A**: in panel (**a**) percentage of correctly found pairs, and percentage of correct pathways – panel (**b**). The percentages of correct pairs: for the collection **B –** panel (**c**), and the collection **C –** panel (**d**). And finally, for the collection **D** of pseudo-real nucleotide sequences sets their correct pairs – panel (**e**), and their pathways – panel (**f**). In all panels sets' sizes are marked by squares (dashed lines) for 200 sequences sets, and by stars (solid lines) for 1000 sequences sets.

We have found that generally the value of ***h*** taken close to the ***½-unit*** distance will produce the highest ratios of correctly reconstructed ancestor–descendant pairs and evolutionary pathways. In **Figure 7** the averages of correctly reconstructed pairs and paths, calculated each time over a collection of QPF translated NJ trees, are shown for different set sizes. It can be seen that in all panels of **Fig. 7** the respective maximums of accuracy are close to the ***½-unit*** distance. However, some fluctuations can be observed, depending on the ratios between the terminal nodes (leaves) to the internal nodes in their respective trees. For the phylogenetic QPF reconstruction of trees obtained from the collection **D** of pseudo-real nucleotide sequences sets, the optimal value of ***h*** (**Figs. 7e** and **7f**) has its maximum shifted slightly into the 0.03 direction – this is due to the different sequences' lengths in the corresponding sets (nevertheless the optimal value is still close to the respective ***½-unit*** distance). What is more, for the larger sets, the percentage of correctly found pairs depends on the threshold ***h*** value, and grows with sequences' lengths – this effect is shown on **Figure 8.** On the other hand, for smaller sets the accuracy of finding correct evolutionary pairs is nearly the same for different ***h*** thresholds. This happens because, for larger sets, trees are disturbed more often than in a case of smaller sets.



a) **b)**

Figure 8. Dependence of the percentage correctly reconstructed ancestor-descendant pairs (panel **a**), and pathways (panel **b**) on the QPF threshold ***h,*** for sets from the collection **A**. To examine the QPF performance, the values of ***h*** *= {0.005; 0.015; 0.025; 0.035; 0.045}* were used.

The overall accuracy of correctly attributed ancestor-descendant pairs and paths is very high, and noteworthy, all the falsely reconstructed pairs for larger sets were found to result from the errors in wrongly generated input trees' topologies, and this can be improved only by using more robust algorithm for their phylogenetic tree construction. Quality of an input tree for the QPF algorithm does have an impact on the final accuracy of the resulting evolutionary pathways.

Conclusions

One of major defects of traditional methods the phylogenetic trees construction is that all molecular sequences are considered as leaves of the tree. Here we have proposed QPF algorithm, which translates a traditional phylogenetic tree, to a tree with all nodes labeled as to their phylogenetic character – in the resulting tree there are no auxiliary nodes, nor there are any nodes' graph-theoretical degree constraints. Translated tree forms an adequate data structure, optimized for a quick evolutionary pathways finding. Therefore, although a resulting, translated tree might lose some information, as a side effect – should there be any missing ancestors (not present in an analyzed sequences set), it is not necessarily a drawback, as for evolutionary pathways analysis only available sequences can be used anyway. The QPF is a robust, novel technique for an unambiguous labeling and analysis of phylogenetic trees generated by any traditional method of user's choice. The assumption that the threshold value ***h*** is optimal when is set to the **½-unit** distance, has confirmed the role of this threshold as acting like a decision classifier distinguishing nodes of traditional phylogenetic tree into two classes – with a parent's, or a child's character. From a practical viewpoint, the proper choice of the ***h*** threshold value

provide well over 95% accuracy, even for larger trees, despite a noise always present in traditionally generated trees.

The QPF algorithm was implemented in C++, and can be obtained upon a written request from the authors. As an input it requires a re-rooted, traditional phylogenetic tree, written in the Newick format, and as an output it generates a QPF translated tree (also in the Newick format), and a reconstructed evolutionary pathways elucidation (saved as a text file).

Acknowledgements

We would like to thank Pat Churchland for looking over the English. This work was partially supported by the EU project SSPE-CT-2006-44405, and also partially supported from the 352/6.PR-UE/2007/7, the 40-10-02/-501-78-44406, and the 40-10-02/501-64-BST-1550 grants.

References

Baum D., 2008. *Reading a phylogenetic tree: The meaning of monophyletic groups*. Nature Education 1 http://www.nature.com/scitable/topicpage/reading-a-phylogenetic-tree-the-meaning-of-41956

Cormen T. H., Leiserson C. E., Rivest R. L., 1989. *Introduction to Algorithms*. MIT Press, Cambridge, MA

Deo N., 1974. *Graph Theory with Applications to Engineering and Computer Science.* Prentice-Hall, Inc.

Felsenstein J., 1978. *Cases in which parsimony and compatibility methods will be positively misleading.* Syst. Zool., 27, 401-410.

Felsenstein, J., 1981. *Evolutionary trees from DNA sequences: a maximum likelihood approach.* J. Mol. Evol. 17, 368–376.

Gusfield D., 1991. *Efficient algorithms for inferring evolutionary trees.* Networks, 21, 19–28.

Hasegawa N., Sugiura W., Shibata J., Matsuda M., Ren F., Tanaka H., 2009. *Inferring within-patient HIV-1 evolutionary dynamics under anti-HIV therapy using serial virus samples with vSPA*, Bioinformatics , 10, 360, doi: 10.1186/1471-2105-10-360.

Kirkpatrick M., Slatkin M., 1993. *Searching for evolutionary pattern in the shape of a phylogenetic tree.* Evolution 47, 1171–1181.

Larget B., Simon D.L., 1999. *Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees.* Molecular Biology and Evolution, 16, 750–759.

Nee S., Homes E. C., May R. M., Harvey P. H., 1994. *Extinction rates can be estimated from molecular phylogenies*. Philos. Trans. R. Soc. Lond. B 344, 77–82.

Pybus O.G., Rambaut A., Holmes E.C., Harvey P.H., 2002. *New inferences from tree shape: numbers of missing taxa and population growth rates.* Syst Biol 51, 881-888.

Rambaut A., 2000. *Estimating the rate of molecular evolution: incorporating non-contemporaneous sequences into maximum likelihood phylogenies*, Bioinformatics 16, 395–399.

Ren F., Ogishima S., Tanaka H., 2003. *Longitudinal phylogenetic tree of within-host viral evolution from noncontemporaneous samples: a distance-based sequential-linking method*, Gene 317, 89–95

Saitou N., Nei M., 1987. *The neighbor-joining method: a new method for reconstructing phylogenetic trees* Mol Biol Evol 4, 406–425.

Slowinski J., Guyer C., 1989. *Testing the stochasticity of patterns of organismal diversity: An improved null model.* Am. Nat. 134, 907–921.