Università di Pisa
Dipartimento di Informatica

Technical Report: TR-01-20

PaTre: a method for Paralogy Trees construction

N. Pisanti[1] R. Marangoni[1,2] P. Ferragina[1]
A. Frangioni[1] F. Luccio[1,2] A. Savona

[1]Department of Computer Science, University of Pisa (Italy)
[2] *CISSC*, University of Pisa (Italy)

Address: Corso Italia 40, 56125 Pisa, Italy.
Tel: +39 050 2212700
Fax: +39 050 2212726

PaTre: a method for Paralogy Trees construction*

N. Pisanti[1] R. Marangoni[1,2] P. Ferragina[1]
A. Frangioni[1] F. Luccio[1,2] A. Savona

[1]Department of Computer Science, University of Pisa (Italy)
[2] *CISSC*, University of Pisa (Italy)

Abstract

Genes belonging to the same organism are called paralogs when they show a significant similarity in the sequences, even if they have a different biological function. It is an emergent biological paradigm that the families of paralogs in a genome derive from a mechanism of iterated gene duplication-with-modification. In order to investigate the evolution of organisms, it can be useful to infer the duplications that have taken place starting from an hypothetical original gene, and that have led to the current paralog genes family. This information can naturally be represented in a *paralogy tree*. Here we present a method, called PaTre, to generate a paralogy tree from a family of paralog genes. PaTre uses new techniques motivated by the specific application. The reliability of the inferential process is tested by means of a simulator that implements different hypotheses on the duplication-with-modification paradigm, and checked on real data.

Keywords: Paralog genes, Paralogy trees, Transformation Distance, Lightest Spanning Arborescence.

Introduction

Biological evolution has generated more and more complex organisms, starting from relatively simple ones. In order to obtain a complex organism, it is necessary to increase the global information contents of a simple genome by means of new genes and new features generations. Several biological mechanisms have been detected as being able to increase this information content in a genome. Some of them, like the cross-transfer, seem to be just random exchanges of genes between different species. Others, like the duplication,

*Research partially supported by the MURST project on "*Algorithms for large data sets.*"

seem to have worked continuously during evolution, as an iterative process.

Since the studies on the yeast genome [8] (which, according to some hypothesis, derives from an ancient quadruplication of an ancestral archeobacterial-like genome), the duplication-with-modification paradigm has been considered to play the major role in the information increase phenomenon [10, 11], at least from a quantitative point of view. According to this paradigm, genes or groups of them (or, in certain cases, the whole genome) are duplicated and thereafter modified, so that the copies eventually produce a slightly different product which plays a new role, thus increasing the available molecular *tools* of the cell. Such a mechanism could explain the presence, in all genomes, of many genes that are very similar to each other, but that generate different proteins. Indeed, current genome representations are usually given by clustering genes into families that show high similarities in their sequences and correlation in their functions. These are named *paralog genes*. Disregarding the possible contribution of other mechanisms, in this paper we assume that paralog genes in a family have been generated by a iterated duplication-with-modification process only. With this assumption, we try to design a method to reconstruct the history of the family in terms of duplication events.

We are not aware of algorithms conceived specifically for this task. Dating gene duplication [3] and building gene trees [12] are related but different works. The former aims to estimate duplication times, which is a more specific goal than ours, and makes use of evolutionary trees, which we do not assume to know. The latter also makes use of assumed phylogenetic information, and gene trees aims to represent relation among ortholog and not just paralog genes. Standard phylogeny reconstruction methods are conceived to answer again a related but different question than the one we address in this paper, and their application for our purpose does not seem to give satisfying results ([14]). Indeed, our approach is close to distance based parsimony methods for phylogeny reconstruction, but as we will see in the next section, its actions are driven from the specific problem, leading to a new *ad hoc* solution for paralogy tree reconstruction. The main novelty stems from the non symmetry of the master-copy relation, as we need a measure to capture the different cost of generating a gene a from another gene b by a duplication of a followed by mutations, rather than the opposite. After this distinction is done, a parsimony principle will drive the choice of a possible master-copy relation among a and b. For this reason, we use the *Transformation Distance* $TD(a, b)$ ([18]), described in section 1.2, that has the property of not being symmetric[1]. From the complete directed graph representing genes as nodes, and weighting the arc (a,b) with $TD(a, b)$, we build

[1] In general $TD(a, b) \neq TD(b, a)$. In this sense the term "distance" is not proper. Nevertheless we will keep on addressing this measure as *transformation distance* as this is how it is known in the literature.

the minimum cost tree. Since the graph is directed, traditional techniques computing a minimum spanning tree cannot be applied here. Therefore, we resort to the *Lightest Spanning Arborescence* problem that we will discuss in section 1.3, and that was introduced in [5].

In the next sections we describe this method, called PaTre (Paralogy Trees), we evaluate it, and we discuss some applications on real data as well as on synthetic one.

1 Background

1.1 The Problem: from paralog genes to paralogy trees

Paralog genes can be defined as *homologous genes in the same organism whose products perform related but not identical functions* [9], and are the results of gene duplications. According to an emerging biological paradigm [10, 11], gene duplication is the predominant mechanism for the evolution of new gene functions: once redundant gene copies are available, mutations may accumulate in one of the copies, thus creating a new gene, while the other one(s) can go on carrying out the original function. From this *duplication with modification* mechanism follows the interest of reconstructing the history of duplications and modifications within a family of paralog genes in an organism. Such a history can be represented as a *directed tree* in which each node represents a paralog gene, and each (oriented) arc represents a duplication event in which the gene placed at the source node has played the role of the template, and the gene placed at the destination node is the (modified) copy. Our goal is to reconstruct the *paralogy tree* from a given family of paralog genes.

1.2 The Transformation "Distance"

The transformation distance (TD for short) has been introduced in [18] as a measure of the cost of transforming a source sequence S into a target sequence T by means of the operations of: *inserting* a new segment, *copying* a segment from S, and *reverse copying* a segment from S. This requires a preprocessing phase to detect common segments between S and T, possibly allowing for some differences among the occurrences of such segments. Therefore, in practice, point mutations that might have taken place in sequences sharing a common ancestor should not affect the detection of similarity among DNA segments. Once the common segments are detected, TD actually measures the minimum weighted chain of segment translocations, insertions, duplications, and reversals that transform a sequence into another. Intuitively, the weight assigned to each one of the operations coincides with its description length. As a consequence, insertion has a high weight, as the whole sequence to be inserted has to be described. The weight

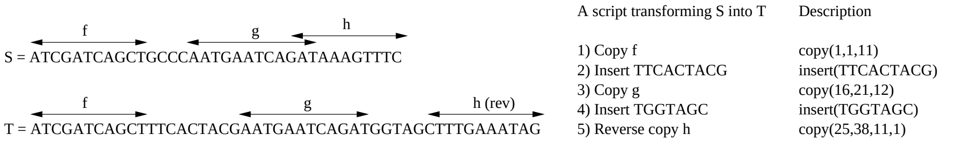

Figure 1: A script that transforms a sequence S into a sequence T, and its description as in [18].

of the copy operation, instead, amounts to specifying the starting position of the fragment in the source sequence, and its length. Finally, the weight of the reverse copy is just one bit more expensive than the normal copy because the reversal information must be given. As an example, in Figure 1 we show a script that spells how a target sequence T is obtained from a source sequence S, with minimum total weight[2]. The figure also shows the description length computed as in [18]. The single operations are coded as in [18], with a code identifying the operation followed by its parameters. The description having minimum length is the transformation distance between S and T. The *copy* operation, for example, is described by its identification code and three parameters (i, j, k), where i is the starting position in T where the segment is copied, j is the position of the fragment in S, and k is the length of the fragment. The reverse copy is similar, plus an additional bit-parameter whose presence indicates the reversal. The description of the insertion operation requires that the sequence to be inserted is given as a parameter. The transformation distance is not symmetric because, in general, a script describing a re-writing of T in terms of S is completely different from one describing S in terms of T. This is basically due to the fact that insertions and deletions are evaluated in a very different way, and thus the simmetry is lost. We will not explain in details how the transformation distance can be computed, but we just recall that this can be done in polynomial time[3] with respect to the length of the sequences. This is done representing all possible scripts as paths on a graph and then finding the shortest path between two suitable nodes. For details, we refer the interested reader to [18].

1.3 The *Lightest Spanning Arborescence* problem

Given a directed graph $\mathcal{G}$ with weighted arcs, and a distinguished node r designated as the root, the *Lightest Spanning Arborescence Problem* asks

[2] We have put identical occurrences of segments to simplify the example. Recall that, in general, point mutations are allowed.

[3] Computing the transformation distance takes $O(n^6)$ time where n is the length of the longest sequence. Note that this is a worst case complexity. In practice, the method is applicable to input up to $100kb$ [18], and over this threshold, it is its space costs that makes it unusable.

for the minimum cost tree rooted in r that spans over all the vertices $\mathcal{G}$. This problem has been solved by Edmond [5] in $O(n^3)$ time, where n is the number of nodes of $\mathcal{G}$. For the case of a complete graph $\mathcal{G}$, a modification of Edmond's algorithm with better practical performances was suggested in [6], and slightly more efficient algorithms have been more recently conceived for the general case [1, 7] as well. However, as we will see later, the tree computation time is not a critical step in PaTre tool because here the number of vertices of $\mathcal{G}$ is at most a few tens. Hence, in our implementation we adopted Edmonds' algorithm with the variation suggested in [6].

2 Our proposal: the PaTre tool

Given a family of n paralog genes $g_1, \ldots, g_n$, in order to construct a paralogy tree from this gene family, our tool takes in input the sequences corresponding to the n genes, and performs the following three main steps:

1. Compute $TD(g_i, g_j)$ and $TD(g_j, g_i)$ for each pair of genes g_i and g_j.
2. Represent the result in a weighted complete directed graph named *TD-graph*.
3. Compute the lightest spanning arborescence on the *TD-graph*.

In our step 1 we compute the transformation distance between every ordered pair of genes belonging to the family of paralogs that we are willing to investigate. Since we have $\Theta(n^2)$ ordered pairs of genes, then we have to compute the distance n^2 times, which makes this phase the most time consuming in our experiment. The distances information can be represented with an $n \times n$ matrix whose cell (i, j) contains the value of $TD(g_i, g_j)$. The same information is represented by step 2 by a complete directed graph, the *TD-graph*, having one node for each gene g_i, and an arc from g_i to g_j weighted $TD(g_i, g_j)$.

For our application on paralogy relations, the transformation distance represents a computationally efficient and biologically significant measure. In fact, we claim that the transformation distance between a gene A and a gene B can quantify the amount of events that would have mutated B to its present form after B was created as a duplication of A (in the case this duplication has taken place). Our belief is biologically motivated as follows. The preliminary detection of (non-exact) common fragments corresponds to detecting which parts of B are conserved with respect to A, even if few point mutations have taken place. The three operations considered in the transformation distance and the corresponding weights ensure that the distance between A and B grows when successive segment reversals, segment insertions, and transpositions take place. In fact, in genome rearrangement literature ([4, 2]) these operations are the most acknowledged candidates for segment level mutations. Concerning the computational complexity, we have that no other distance that takes into account these events results in a computationally tractable problem.

Once the *TD-graph* is built, in step 3 we apply the algorithm that finds the lightest arborescence algorithm on it. We cannot use traditional algorithms solving the minimum spanning tree problem because our distance is not symmetrical, and therefore the resulting *TD-graph* is directed. For the same reason, traditional phylogeny tree inference methods based on distance matrices cannot be of any use here in order to build a tree. The non symmetry has a biological meaning in this specific problem. Namely, the fact that the minimum cost tree includes the arc from gene A to gene B (and therefore not the one from B to A), implies that gene B is the result of a past duplication of gene A. Given two sequences A and B, the difference between $TD(A, B)$ and $TD(B, A)$ mainly depends on non-common fragments. In particular if A contains many fragments that do not appear in B, and the opposite holds less (in other words, from A to B there are more deletions than insertions), then we have $TD(A, B) < TD(B, A)$. Therefore, it is more likely that the parsimony criteria will lead to a resulting paralogy tree where B will be a descendant of A rather than the opposite. This corresponds to a choice that acknowledges deletions as more probable than insertions, after a gene is duplicated. This complies with the biological paradigm called *duplication with loss*, based on the observations that losses do take place after a duplication, while insertions are more likely to occur independently from paralogy relations as they are more frequently due to *horizontal events* such as crossing over.

Since there is one minimum weight arborescence for each possible fixed root, the computation in step 3 should be repeated $\Theta(n)$ times, one for each possible gene as a root, to find the arborescence having minimum weight. This would result in a $O(n^4)$ time complexity algorithm using Edmonds' approach [5]. It should be pointed again that the value of n is the cardinality of the paralogs gene family, which hardly is more than a few tens genes. This is why the computation of the lightest arborescence tree is not computationally critical. Nevertheless, in order to avoid the $O(n^4)$ time cost, we add a dummy node r to the *TD-graph*, having n arcs with very high weights leaving it, each one reaching one of the other n nodes, and no arc entering it. The minimum weight spanning arborescence is computed using r as a root, and the resulting tree will contain only one arc among those leaving r, due to the high weights given to these arcs. Therefore, we can just designate the tail of that arc as the root of the paralogy tree, and the tree obtained by pruning the dummy node is exactly the minimum weight arborescence of the TD-graph. Hence, a unique rooted tree is the result in our method, and this is computed in $O(n^3)$ time. This tree is the one we suggest as a paralogy tree for the gene family under investigation. In sections 3 and 4 we will validate this assumption first on synthetic data, and then on real data.

3 Experimental results based on synthetic data

In order to have a first experimental assessment of the soundness of our method, the initial tests have been carried out on synthetic data, whose expected optimal tree is already known (as it is the output of a simulator). The simulator is described in section 3.1, and a validation of it is given in section 3.2. The experimental results are presented in Section 3.3.

3.1 The simulator

Our *simulator of evolution* generates a paralogy tree of a gene family starting from an arbitrary initial sequence. Starting from a given unique sequence, the simulator performs duplications and mutations according to the following model. A new sequence s_1 can be generated from a sequence s_0 because a few among the following events take place:

1. A fragment of s_0 is inserted in s_1.
2. A fragment of s_0 is inserted reversed in s_1.
3. Single bases are added or modified in s_1.
4. Fragments are inserted in s_1.

Notice that the insertion of fragments from sources other than s_0 corresponds to biological events such as horizontal transfers, which are segment based mutations that are *external* to paralogy relations. This is why in our simulator such events are introduced as random insertions. The simulator implements this model in several steps. First, a random number of random points are selected in s_0, thus actually defining fragments of s_0. Second, a random number of random sequences are generated. Third, a new sequence is generated using fragments of s_0 as they are selected at the first step, and the random ones generated at the second step. In this phase, the fragments have 50% probability to be reversed. Moreover, once s_1 is built, random point mutations are performed. This process is then iterated after a new sequence playing the role of the s_0 (the template) is selected. These operations correspond to applying random insertions, reversal, and transposition on fragments, and random point mutations on all sequences. Our tool allows two possible ways to choose at each iteration the new sequence that is a template. The first way just uses an uniform probability, that is all genes have the same probability to be selected as a template for a new duplication. Otherwise, we can assume that in the evolution, the probability of selecting a gene as a template for a duplication decreases each time that the gene behaves as a template. In this way, the *youngest* genes have a higher probability to generate a duplicate. This strategy is motivated by a qualitative observation that the trees generated with this probability distribution are more similar to those reconstructed on real biological sequences. In any case, this is not relevant for the present study, and it will be matter of the

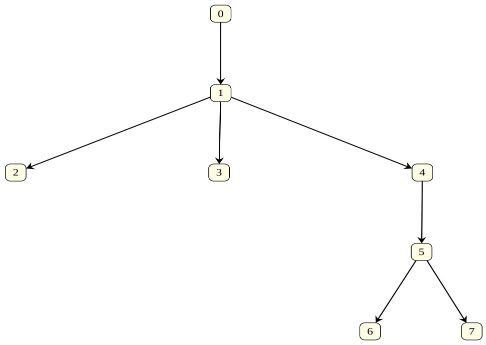

Figure 2: Example of a simulator output. The labels assigned to the nodes show the order of generation of the corresponding genes.

future development of our work.

To test PaTre on simulated data we first apply our simulator to a randomly generated initial gene, thus obtaining an arborescence $\mathcal{A}$ like the one shown in Figure 2. Hence, we feed PaTre with the paralog genes family consisting of the sequences associated with the nodes of $\mathcal{A}$. What we expect then as a result from PaTre is exactly $\mathcal{A}$, formerly obtained from the simulator.

3.2 Validating the simulator

In order to validate the reliability of the simulated generation of paralog genes, we have designed a test which uses ClustalW[4], a clustering algorithm. The tests work as follows. Chosen a family of paralogs, we randomly extract one of the genes and we give it as an input to the simulator, to which we ask to generate as many paralogs as the number of real genes in the family. Afterwords, we collect together real and simulated genes, and run ClustalW. Intuitively, the more the simulator is faithful to the real paralog genes generation, and more it has generated genes that are similar to those actually belonging to the family. In this case, what we expect from ClustalW is an output that exactly pairs one real and one simulated gene. If this happens, we say that a *perfectly mix* has taken place. More in general, we define a mixing index μ that measures how much real and simulated data have mixed, and that we compute as follows. Let k be the number of internal nodes (that is without the leaves and the root) in the tree output

[4]ClustalW ([16]) is a widely used tool for multiple alignment that, among other facilities, groups some given input sequences by means of the results of their pairwise comparisons (using the method known as *neighbor joining*). The output clustering is given as a tree. The version we used is the one hosted by EBI, the european bioinformatic institute (http://www.ebi.ac.uk/clustalw/).

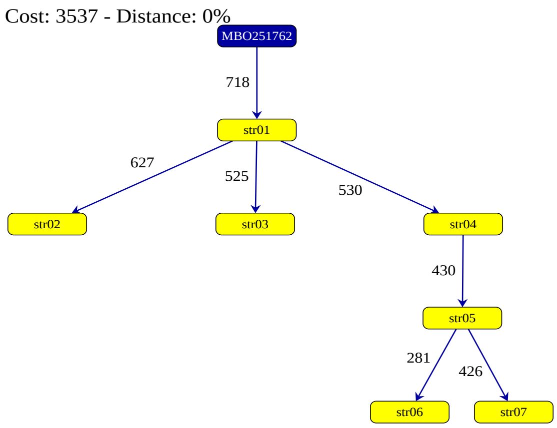

Figure 3: The paralogy tree for the complete input set.

by ClustalW. We set $\mu = k'/k$, where k' is the number of internal nodes that have both real and simulated genes as descendants. It is worth remarking that there is a large variety of families that show a μ index of 100%, that is real and simulated sequences are completely mixed with each others. This means a good *mimetic* capability of the simulated data with respect to real one. As a side result of this test, it is interesting to notice that the value of the parameter μ remains about constant when we explore the same family in different organisms, even if they are very far in the evolution.

3.3 Results on synthetic data

We have used paralog gene families generated by the simulator as input data for PaTre. The results show that, on these data, PaTre is *always* able to reconstruct correctly the paralogy tree generated by the simulator when the input set contains all the sequences belonging to the history of the ancestral sequence. As an example, Figure 2 shows the simulated evolution of a family, and Figure 3 shows the tree reconstructed by PaTre with the sequences generated by the simulator as input. One can see that the two trees are identical. This type of result is achieved independently of the probability function used for the choice of the master for a new duplication.

On the other hand, when the input set does not contain all the sequences generated starting from the common ancestor, or when there are sequences not belonging to the history of the family, the method gives as optimal solution an arborescence that in general may be rather different from the expected one. In fact, if a sequence in the input set is missing, one could expect a resulting tree that basically coincides with the tree generated with complete data, except around where the missing node was supposed to be

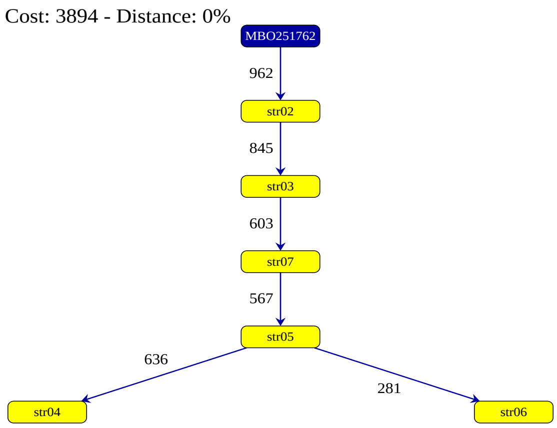

Figure 4: The paralogy tree for the input set where the sequence $str01$ is missing.

placed. PaTre, instead, generates a tree which has a topology that may look very far from the one of the complete tree (compare Figure 3, and Figure 4). This problem cannot be ignored as in real data it is unfortunately very probable that there are some missing genes with respect to the whole paralogy family. This can happen either because the sequencing of those genes has not taken place yet, or because it has erroneously been classified in another paralogy family. In may also be the case that a gene is not detected simply because it does no longer exist. This is actually very common, but in most of the cases such genes did not even ever duplicated, and they eventually disappear. In these cases, the gene would have been a leaf in the paralogy tree, and hence its absence has no consequence on the tree. The interesting cases are indeed those in which the gene is missing because it has not been classified as belonging to its correct paralogy family.

Exploring the set of sub-optimal solutions, we found out that when the expected tree is not the optimal due to incompleteness of data, PaTre includes the expected tree among the best solutions, usually with a *small difference* in the cost with respect to the one detected as optimal. On the other hand, when PaTre receives in input the complete set of sequences, the solution returned by PaTre is not only the correct one, but it has usually a significantly lower cost with respect to the second best solution. This observation suggest us to investigate the possible usage of the distance between optimal and sub-optimal solutions as a starting point to the building of a tool able to measure the incompleteness degree of a family of paralogs. Nevertheless, we also have to pay attention to cases of a sort os unavoidable uncompleteness. Namely, when the family presents very ancient duplications, it may be the case that older ancestors mutated so much since then, that it is no longer possible to detect them as those playing the role that they actually plaied. Therefore, in some sense, the gap between the value of the optimum and the one of the second best solution can also be intended as a measure of the robusteness of the result output by PaTre. Computing an approximation of the gap between the optimal solution and the second best one, can be done very efficiently since Edmond's algorithm is capable of computing in $O(n^2)$ time, once the optimal tree is known, the *reduced costs* of the arcs. The reduced cost of an arc e gives a lower bound on the additional value that the optimal solution would have if the arc e was forced to belong to it. Therefore, all arcs belonging to the optimum have reduced cost equal to zero, and the smallest non-negative reduced cost gives us a lower bound on the additional cost of the second best solution, and hence on the gap between the optimal and the best sub-optimal. If this lower bound is already larger than a given threshold, say 5% of the optimum, then according to out hypothesis, this is enough to assume that the family is reasonably complete, and the resulting tree is a robust solution. Otherwise, a proper set of *good* solutions has to be found in order to compute the exact distance between the optimum and the second best solution. To reach this purpose, PaTre offers the possibility of performing suitable extra runnings of Edmond's algorithms in the following way. For all edges e belonging to the optimal solution, the minimum weight arborescence is re-computed forcing the absence of e. This step can be implemented by assigning a very high weight to the arc e, and re-computing the minimum weight tree. This extra runs take $O(n^4)$ time because one $O(n^3)$ time algorithm is applied $\Theta(n)$ times (once for each possible edge whose absence is forced). Although this cost looks high, we remind that this is feasible as n in never too large. We have:

Proposition 1 The second best solution is always found in the extra runs.

Proof. Let T' be the optimal solution, and T'' be the second best one. We have that T'' necessarily differs from T' in at least one edge, that is, there exists $e \in T'$ such that $e \notin T''$. Therefore, when this e is forced to be missing we will find T'' as a new optimal solution. □

4 Preliminary experimental results on real data

Before applying PaTre to real biological data, we would like to point out that in biology there are no direct evidences concerning neither the history of a family of paralogs, nor about the loss of genes from a family. Therefore, the main evaluation of our method remains the test on simulated data. On the other hand, it is useful to discuss the results obtained by applying PaTre on real sequences, in order to test the reliability of the detected duplication relationships, and the possibility of extending this approach from lower to higher organisms.

Following these considerations, we paid attention to the cases in which there are some evidences concerning that paralog genes come from duplication processes, or to the genomes in which comparative studies have generated some suggestions about the history of a family of paralogs.

4.1 Some studies on a family of *Mycoplasma pneumoniae*

The *Mycoplasma pneumoniae* is a bacteria whose complete genome sequence is available for a few years already. We chose the *heat shock protein* family within the genome of this familym, which contains eight genes. It should be complete because the genome has been sequenced completely and estensively studied. Moreover, being this a bacteria, it is a relatively little developed organism, and hence probably all ancestors of existing genes of this family are still present in the genome, even if mutated.

The tree generated by PaTre is shown in Figure 4.1. The nodes are labeled with the name of the genes. The number that ends the name of the gene is related to the position of the gene, in the sense that consecutive numbers refer to adjacent genes. Therefore, we can notice that there are two adjacent genes in the family, as well as a group of four consecutive ones. These are very likely duplicates of eachother because duplications can happen (but this is not necessarily always the case) *in tandem*, that is the copy is placed right next to the template. The PaTre tool does not have this information, that is there is no phase in which it uses informations regarding the position of the genes. Nevertheless, it does locate these genes close to eachother in the tree. Therefore, we have that PaTre detects these suspected tandem duplications mentioned above. Finally, we have that the output tree is the only solution in a wide range (15%) around the value of the optimum, which would confirm the completeness of the family.

4.2 A case of a higher organisms: the plant *Arabidopsis thaliana*

The one of *Arabidopsis thaliana* is the only complete genome of a plant available so far[5]. Like all plants, it is an highly developed organisms and, therefore, it is not the most suitable one for applying PaTre. In fact, due to the high age of the genes, it is unlikely that a current paralog genes family is complete. That is, common ancestors of present time paralog genes might no longer exist. As a result, the paralogy tree otput by PaTre could hardly recover the actual relations, especially concerning the highest part of the tree. Nevertheless, we applied PaTre to a family of paralogs of this genome for several reasons. First, for this family we do know studies

[5]The complete sequence is available at http://www.arabidopsis.org.

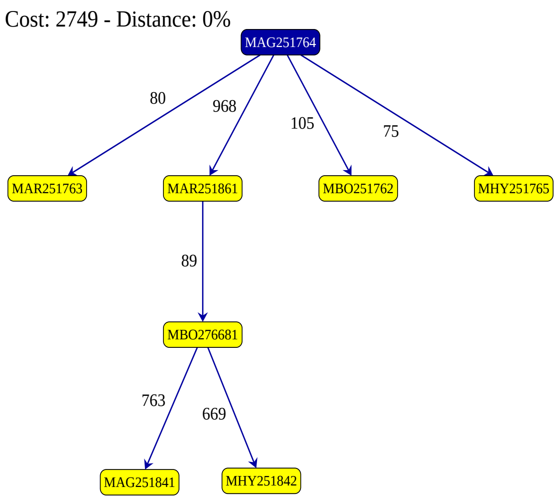

Figure 5: The paralogy tree output by PaTre for the 8 genes of the heat shock protein family of *Mycoplasma pneumoniae*.

that formulate hypothesis of past duplication events and hierarchical relation among presently existing genes ([14]), and therefore we know what to expect as reasonable results. Second, we wanted to check whether, with this very probably incomplete family as an input, the result is indeed a solution whose value is very close to other sub-solutions.

4.2.1 The SHAGGY/GSK-3 family

The Shaggy like kinases family in *Arabidopsis thaliana* (AtSK, see [17, 15]) is actually a sub-family of the GSK-3 one. The biochemical role of the proteins encoded by these gene is widely studied in mammals, while their function in plants is less clear. Nevertheless, it is known that the biological role of these protein is very variable, and that this family seems to be very ancient. In order to minimize the noise due to high divergence, we used the catalytic domain sequences, which are relatively highly conserved. At present time, in *Arabidopsis thaliana* ten genes have been classified into this family. Therefore, we used for our applications ten genes, which are divided into four groups: AtSK-alpha, AtSK-gamma, and AtSK-epsilon of group I, AtSK-eta, AtSK-dzeta, and AtSK-iota of group II, AtSK-beta and AtSK-theta of group III, and finally AtSK-kappa and AtSK-delta of group IV. The partition in groups corresponds to acknowledged sequence similarities. These ten genes are arranged in five different chromosomes in *Arabidopsis thaliana*. Previous studies on this family (see [14] for a review) have suggested models of its evolution that include the following duplications

· In group II, AtSK-dzeta is ancestor of AtSK-iota by translocation of a fragment.
· In group II, AtSK-dzeta is also ancestor of AtSK-eta, by another translocation of another fragment.
·Within group I, the AtSK-alpha is an ancestor of AtSK-gamma, again by translocation.

These assumptions are motivated by the observation of high similarities in the sequence and by the fact that in AtSK-dzeta (resp. AtSK-alpha) there is an additional intron with respect to AtSK-eta (resp. AtSK-gamma), which can be justified with a translocation event that moves a fragment and this splits a coding fragment into two. The application of phylogeny reconstruction based on distances among the amino acid sequences of the corresponding proteins, also suggests the following possible divergence by duplications, although the direction cannot be specified, as the common ancestor may not be any of the existing genes.

· Genes AtSK-delta and AtSK-kappa (that is, group IV).
· Genes AtSK-beta and AtSK-theta (group III).
· Gene AtSK-gamma related to AtSK-epsilon, while the results above only mention the relation of the former with AtSK-alpha. These three genes compose group I.

Since we assume that the family is not complete, and in particular the earliest ancestors are probably missing, our goal in this specific application of PaTre is not of recovering the paralogy tree, but rather to detect the most possible of the known (or, actually, supposed) past duplications events.

4.2.2 The paralogy tree output by PaTre

First of all, we have measured the mixing rate between real data of the SHAGGY/GSK-3 family, and the one obtained by the simulator starting with one of the ten genes as initial sequence. Since the evolution has taken place not on the catalytic domain only, but rather on the whole gene sequence including introns, we have performed the mixing and the run of ClustalW on the complete genes sequences. The result is a 100% mixing rate. Secondly, we have applied PaTre giving as an input the catalytic domain of the DNA sequences of the genes. Figure 4.2.2 shows the paralogy tree output by PaTre where the edges are labeled with the transformation distance between the source node and the sink one. We can immediately see that the four groups are well separated in the obtained tree. More into detail, *all* the assumed duplications are represented in the tree as the corresponding genes are connected by a direct edges. Moreover AtSK-dzeta indeed results in an ancestor of AtSK-iota, as well as of AtSK-eta, as ex-

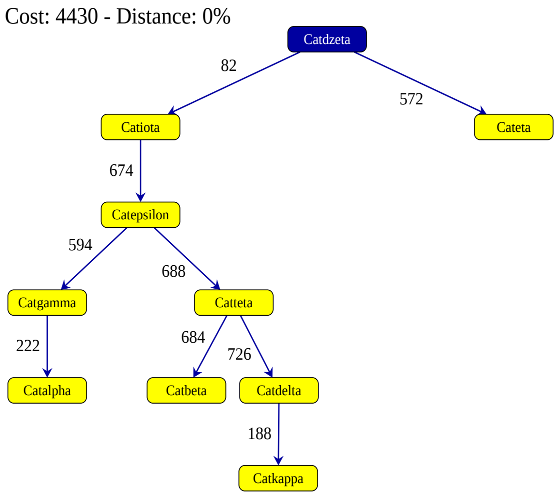

Figure 6: The paralogy tree output by PaTre for the catalytic domain sequences of the SHAGGY/GSK-3 family in *Arabidopsis thaliana*. The labels on the edges represent the transformation distances.

pected. The only difference with the assumptions is that AtSK-gamma is an ancestor of AtSK-alpha rather than the other way around. This can be explained by the fact that AtSK-gamma results acknowledged ad a duplication of AtSK-epsilon (as expected), and eventually this results incompatible with an edge directed from AtSK-alpha to AtSK-gamma. Within the range of 5% of the optimal tree (that costs 4430), there are in total 19 solutions. This would confirm the non completeness of the family, in the sense that ancestors have disappeared.

Considering that *Arabidopsis thaliana* is a highly developed organism, the results we obtained for this family are satisfying. The division into the known groups is clear. All expected relations are detected, and in all cases but one the expected orientation to edges is also given.

Conclusions and further work

The most immediate further work is to find data concerning sequenced paralog genes families in order to apply PaTre and check whether the expected results are given. This is still in the direction of testing the tool. Independently from that, we plan to apply PaTre to other families that are complete (or are assumed to be so). Such applications could also further explore the possibility to use PaTre as a tool to measure the incompleteness of paralogs families. In particular, using the optimum-suboptimum distance as a quantitative parameter of the incompleteness of the input set.

Another possibility is to use a "distance" alternative to the *transformation distance*. This would help also in decreasing the computational complexity of the tool, which could lead to applicability to larger data sets such as entire genomes.

As a further step of this project we aim to compare paralogy trees of different families, both in the same genome and in different ones. In particular, as a new way of performing genome comparisons and phylogeny reconstruction, it could be interesting to compare the shape (i.e., some topological properties) of the paralogy tree of the same families in different organisms. For example, we could check how much certain organisms tend to have deeper trees, with respect to others, for the same family. With respect to traditional phylogenetic studies, instead of comparing single genes belonging to different organisms, we could compare entire paralogy trees for a given family of paralog genes in different species. This approach could avoid many problems related to the choice of the gene and the relative cross-similarity between orthologs and paralogs. Moreover, it could give information about the strategy of new genes creation, and it could address the question posed by the simulator structure, that is whether the template for a new copy should be chosen among all the existent paralogs with uniform probability, or, on the contrary, the more recent genes should have a higher probability to be chosen. In the latter case, it would be possible to infer the existence of a *timing* system on the DNA, which marks in different ways ancient and recent paralogs. Finally, within the same genome, it could be interesting to compare the shapes of different trees related to different families. This could help in current studies of the different duplication rates of genes.

Acknowledgments

We thank Dr. Raquel Tavares (Laboratoire d'Algorithmique Combinatoire, Institute Pasteur) for the catalytic domain sequences of the SHAGGY/GSK-3 family in *Arabidopsis thaliana*, and for pointing us some papers on this family. We are also grateful to Chiara Pisanelli for taking care of running PaTre on some genes families.

References

[1] M. Adler and M. Mitzenmacher. Towards compressing web graphs. In *IEEE Data Compression Conference*, 2001.

[2] M. Blanchette, T. Kunisawa, and D. Sankoff. Parametric genome rearrangement. *Gene-Combis*, 172:11–17, 1996.

[3] K. Chen, D. Durand, and M. Farach-Colton. Notung: Dating Gene Duplication using Gene Family Trees. *Proceedings of RECOMB 2000*, pages 96–106, 2000.

[4] D.A. Christie. *Genome Rearrangement Problem.* PhD thesis, University of Glasgow, 1998.

[5] J. Edmonds. Optimum branching. *J. Res. Nat. Bur. Standards*, 71B:233–240, 1997.

[6] M. Fisichetti and P. Toth. An efficient algorithm for the min-sum arborescence problem on complete digraphs. *SIAM Journal of Computing*, 5:462–434, 1993.

[7] H.N. Gabow, Z. Galil, T. Spencer, and R.E. Tarjan. Efficient Algorithms for Finding Minimum Spanning Trees in Undirected and Directed Graphs. *Combinatorica*, 6(2):109–122, 1986.

[8] P.E. Hansche. Gene duplication as a mechanism of genetic adaptation in *Saccharomyces Cerevisiae*. *Genetics*, 79:661–674, 1975.

[9] J.C.D. Hinton. The *Escherichia coli* genome sequence: the end of an era or the start of the FUN? *Molecular Microbiology*, 26(3):417–422, 1997.

[10] S. Ohno. *Evolution by Gene Duplication.* Springer-Verlag, 1970.

[11] T. Ohta. Simulating Evolution by Gene Duplication. *Genetics*, 115:207–213, 1987.

[12] R.D.M. Page. Gene Tree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14:819–820, 1998.

[13] J.C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology.* PWS Publishing Company, 1997.

[14] R. Tavares. *Contribution à la caractérisation de la sous-famille des protéines sérine/thréonine kinases du type SHAGGY/GSK-3 chez Arabidopsis thaliana.* PhD thesis, University of Paris sud, 2000.

[15] R. Tavares, S. Aubourg, A. Lecharny, and M. Kreis. Organization and structural evolution of four multigene families in *Arabidopsis thaliana*: AtLCAD, AtLGT,AtMYST and AtHD-GL2. *Plant Molecular Biology*, 42:703–717, 2000.

[16] J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTALW: Improving the Sensitivity of Pregressive Multiple Sequence Alignment through Sequence Weighting, Position Specific Gap Penalties and Weight Matrix Choice. *Nucleic Acid Research*, 22:4673–4680, 1994.

[17] G. Tichtinsky, R. Tavares, A. Takvorian, N. Schwebel-Duguè, D. Twell, and M. Kreis. An evolutionary conserved group of plant gsk-3/shaggy-like protein kinase genes preferentially expressed in developing pollen. *Biochim. Biophys. Acta*, 1442:261–273, 1998.

[18] J.S. Varrè, J.P. Delahaye, and E. Rivals. Transformation Distances: a family of dissimilarity measures based on movements of segments. *Bioinformatics*, 15:194–202, 1999. Also presented at GCB98.