

Fast NJ-like algorithms to deal with incomplete distance matrices

Alexis Criscuolo^{1,2,3} and Olivier Gascuel^{1*}

¹Equipe Méthodes et Algorithmes pour la Bioinformatique

LIRMM, CNRS - Université Montpellier 2

161 rue Ada, 34392 Montpellier Cedex 05, France

<http://www.lirmm.fr/mab>

²Groupe Phylogénie Moléculaire

ISEM, CNRS - Université Montpellier 2

C.C. 064, 34095 Montpellier Cedex 05, France

³Equipe Bioinformatique Théorique,

LSIT, Université Louis Pasteur, Strasbourg 1,

Pôle API, Boulevard Sébastien Brant,

BP 10413, 67412 Illkirch Cedex, France

Emails:

AC: criscuol@lirmm.fr

OG: gascuel@lirmm.fr

*Corresponding author

Abstract

Background

Distance-based phylogeny inference methods first estimate evolutionary distances between every pair of taxa, then build a tree from the so-obtained distance matrix. These methods are fast and fairly accurate. However, they hardly deal with incomplete distance matrices. Such matrices are frequent with recent multi-gene studies, when two species do not share any gene in analyzed data. The few existing algorithms to infer trees with satisfying accuracy from incomplete distance matrices have time complexity in $O(n^4)$ or more, where n is the number of taxa, which precludes large scale studies. Agglomerative distance algorithms (e.g. NJ [1,2]) are much faster, with time complexity in $O(n^3)$, which allows huge datasets and heavy bootstrap analyses to be dealt with. These algorithms proceed in three steps: (a) search for the taxon pair to be agglomerated, (b) estimate the lengths of the two so-created branches, (c) reduce the distance matrix and return to (a) until the tree is fully resolved. But available agglomerative algorithms cannot deal with incomplete matrices.

Results

We propose an adaptation to incomplete matrices of three agglomerative algorithms, namely NJ, BIONJ [3] and MVR [4]. Our adaptation generalizes to incomplete matrices the taxon pair selection criterion of NJ (also used by BIONJ and MVR), and combines this generalized criterion with that of ADDTREE [5]. Steps (b) and (c) are also modified, but $O(n^3)$ time complexity is kept. The performance of these new algorithms is studied with large scale simulations, which mimic multi-gene phylogenomic datasets. Our new algorithms — named NJ*, BIONJ* and MVR* — infer phylogenetic trees that are as least as accurate as those inferred by other available methods, but with much faster running times. MVR* presents the best overall performance. This algorithm accounts for the variance of the pairwise evolutionary distance estimates, and is well suited for multi-gene studies where some distances are accurately estimated using numerous genes, whereas others are poorly

estimated (or not estimated) due to the low number (absence) of sequenced genes being shared by both species.

Conclusions

Our distance-based agglomerative algorithms NJ*, BIONJ* and MVR* are fast and accurate, and should be quite useful for large scale phylogenomic studies. When combined with the SDM method [6] to estimate a distance matrix from multiple genes, they offer a relevant alternative to usual supertree techniques (e.g. [7]). Binaries and all simulated data are downloadable from [8].

Background

Phylogeny inference methods can be classified into two main categories: character-based (e.g. maximum-parsimony or maximum-likelihood) and distance-based approaches. The latter have low running times which are quite useful (mandatory in some cases) to perform large-scale studies and bootstrap analyses. A number of computer simulations (e.g. [9-17]) have shown that distance methods are fairly accurate, though not as accurate as likelihood-based methods that are much more time consuming. Using any distance-based method first requires to estimate the pairwise evolutionary distances between every taxon pair. These distances are usually estimated from DNA, RNA or protein sequences, but can also be obtained from DNA-DNA hybridization experiments or, e.g., computed from morphological data (see [18] for a review on distance estimation from various data types).

In the last few years, phylogenomic studies (i.e. phylogeny reconstruction from large gene collections [7]) have instigated to the development of fast tree-building techniques being able to infer trees from datasets comprising hundreds of genes and taxa. The *low-level gene combination* involves concatenating the different genes into a unique *supermatrix of characters*, and then analyzing this matrix with a standard tree building method. This approach was shown to perform poorly when combined with distance methods, due to inaccurate distance estimations from such large heterogeneous character matrix [6]. Better distance-based trees are obtained by extracting the phylogenetic information from each gene separately, and then combining resulting information sources into a unique *distance supermatrix*. The *Average Consensus Supertree* (ACS [19]) and *Super Distance Matrix* (SDM [6]) techniques input a collection of distance matrices being estimated from each gene separately (the so-called *medium-level combination*), or being equivalent to the gene trees (the *high-level combination*). These distance matrices are deformed, without modifying their topological message, and then averaged to obtain the distance supermatrix, which is finally analyzed using a distance-based tree building algorithm.

Estimating the distance supermatrix is fast. However, missing entries may occur in distance supermatrices depending on the extent of taxon overlap within the

source matrices. For example, with the two large data sets of Driskell et al. [20], which were collected from Swiss-Prot and Gen-Bank thanks to a computer program, the ratio of missing distances is ~19% and ~1.2%, respectively. These distances are missing because only a few genes are sequenced within each species, meaning that a number of species pairs do not share any sequenced gene in common and cannot be compared using available data. However, Driskell et al. showed that, despite the sparseness of data and the fact that only a small subset of these data is potentially phylogenetically informative, a topological signal still emerges, which provides useful insights into the tree of life (see [20] and below for details). Analogous findings were reported by a number of authors in various contexts (e.g. [21-23]), and tree building from sparse data has become topical, as can be seen from the flourishing literature on supertrees.

However, tree building from incomplete distance matrices is NP-hard [24], and thus practical algorithms are heuristics. The indirect approach involves first estimating missing distances by applying an ultrametric [25], additive [26], decomposition-based [27], or quartet-based [28] completion algorithm. The TREX package [29] provides several implementations of such algorithms to be used before tree building using any standard method with the completed matrix. The direct approach involves using a weighted least-squares (WLS) algorithm and associating missing distances with null weight (i.e. infinite variance), which means that missing distances are simply discarded from WLS computations [18, pp. 449]. The FITCH algorithm [30] from the PHYLIP package [31] and the MWMODIF algorithm [32] from TREX implement this technique. A combination of both direct and indirect methods is provided by MW* [33] (also available in TREX); this algorithm first applies an ultrametric or additive completion algorithm (depending on the density of missing distances) and then infers a tree using MWMODIF, where weights are set to 1.0 for known distances, 0.5 for estimated distances, and 0.0 for missing distances (if any remain). All these (direct or indirect) algorithms have $O(n^4)$ time complexity or more, where n is the number of taxa. This limits their application to medium-sized datasets (say 200 taxa without bootstrapping, see below).

Agglomerative algorithms are much faster and allow dealing with thousands of taxa, as soon as the distance matrix is complete. The most popular of them is the Neighbor-Joining (NJ) algorithm [1,2]. Starting from a star tree, agglomerative algorithms iteratively perform the three following steps, until the tree is completely resolved:

- (a) select a taxon pair xy that is agglomerated into a new node u ;
- (b) estimate the length of the two so-created external branches ux and uy ;
- (c) replace x and y by u in the distance matrix, and estimate the new distances between u and the not-yet-agglomerated taxa.

Step (a) is more time consuming than the two other steps, because of testing all the $O(n^2)$ taxon pairs to select the optimal one. To this purpose, NJ optimizes a numerical criterion that is denoted as Q_{xy} . This criterion admits several interpretations related to the Minimum Evolution principle [1,34], but also to the acentrality of the considered pair [35,36]. In this last interpretation (used here), Q_{xy} measures how much the path joining x to y is far from the other taxa $i \neq x, y$. The xy pair maximizing Q_{xy} corresponds to the two taxa which are most distant from the other ones and is the best candidate for agglomeration. Another criterion, denoted as N_{xy} , is used by ADDTREE [5]; this second criterion is based on the four point condition [37,38] and counts the number of taxon quartets $xyij$ where x and y are neighbors. When the distance matrix exactly corresponds to a tree (it is then said to be *additive*), N_{xy} indicates all pairs of sibling taxa in the tree, whereas Q_{xy} indicates just one such taxon pair. We shall see that this property of N_{xy} is a great advantage when dealing with incomplete distance matrices. Indeed, Q_{xy} is sometimes unusable whereas N_{xy} is still informative.

Steps (b) and (c) essentially correspond to distance averaging, which requires $O(n)$ run time. These three steps being repeated $n-2$ times, agglomerative algorithms require $O(n^3)$ time when using the Q_{xy} pair selection criterion, and $O(n^4)$ with N_{xy} [39].

Several refinements of the NJ algorithm have been proposed. BIONJ [3] minimizes the variances associated to the new distances being estimated during each

reduction step (c). This way, BIONJ makes use at each iteration of reliable distance estimates to select the new taxon pairs to be agglomerated. To this aim, BIONJ uses a simple Poisson model of the variances and covariances of the distances being contained in the initial distance matrix. BIONJ was generalized into the *Minimum Variance Reduction* algorithm (MVR [4]), a WLS variant of which can deal with any distance variance model, but which does not account for the distance covariances. It has been shown using computer simulations that this variant (named WLS-MVR in [4] but referred here as MVR for simplicity) has similar accuracy as NJ when applied to distance matrices estimated from one-gene alignments [4]. WEIGHBOR [40] further refines BIONJ approach and uses an agglomeration criterion which accounts for the variances of evolutionary distances. All these algorithms require $O(n^3)$ time. Faster, sophisticated distance-based algorithms have been proposed in the last few years (e.g. [41-46]), some of them being clearly more accurate than NJ and BIONJ (e.g. FASTME [42] and STC [44], in $O(n^2 \log(n))$ and $O(n^2)$, respectively).

In this paper, we propose an adaptation of the agglomerative scheme to quickly infer phylogenetic trees from incomplete distance matrices. We show that the Q_{xy} criterion may be rewritten to express the mean acentrality of the xy taxon pair. In the same way, the N_{xy} criterion may be rewritten to express the mean number of taxon quartets where x and y are neighbors. By estimating these two means using all available (non-missing) distances, we define the two criteria Q_{xy}^* and N_{xy}^* which allow for the selection of taxon pairs in step (a), even when the distance matrix is incomplete. Using these two new criteria in the agglomerative scheme requires $O(n^3)$ and $O(n^4)$ run time, respectively. A limitation of Q_{xy}^* and N_{xy}^* is that they cannot be computed when the distance corresponding to the xy pair is missing (see Methods for more). However, this difficulty is inherent to the problem of building trees from incomplete distance matrices and is encountered (in various forms) by all methods to deal with this problem. Moreover, N_{xy}^* partly circumvents this difficulty thanks to its ability to indicate several relevant pairs, rather than a single one with Q_{xy}^* (see Methods for more). As running N_{xy}^* requires $O(n^4)$ time, we use a filtering technique: at each step (a) we use Q_{xy}^* to select the s most promising pairs for agglomeration, and then use N_{xy}^* to select the best of these s pairs. This computational

trick (and other refinements, see Methods) greatly improves the accuracy compared to using Q_{xy}^* only, while requiring $O(sn^3)$ time, where s is a small constant ($s = 15$ in our experiments). Finally, the original NJ, BIONJ and MVR formulae corresponding to steps (b) and (c) essentially are distance averaging and are easily adapted to incomplete matrices. The three new algorithms are named NJ*, BIONJ* and MVR*, respectively.

Results and Discussion

Several computer simulations are presented in this section to assess the performance of NJ*, BIONJ* and MVR*. We first compare the agglomeration criteria Q_{xy}^* , N_{xy}^* and their combination with distance matrices that are additive, but contain missing entries. Then, using more realistic datasets, we compare NJ*, BIONJ*, MVR* to FITCH [30] and MW* [33], in terms of both topological accuracy and run times.

Comparison of agglomeration criteria

Our approach is similar to Makarenkov and Lapointe's [33]. We analyze with various algorithms and criteria a distance matrix with randomly deleted entries. The distance matrix we use is additive, i.e. is obtained from a tree by computing the path length distance between every taxon pair. Let T denote this tree and (T_{ij}) be the corresponding distance matrix, where T_{ij} is the path-length (or patristic) distance between taxa i and j in T . When no entry is missing, such an additive matrix uniquely defines T , which is recovered by any consistent algorithms (as are all algorithms being tested here). When entries are missing in (T_{ij}) , recovering T becomes a difficult task (see above), and we measure how well the algorithms perform when given an increasing number of missing distances. Such data thus are not realistic from a biological stand point, as evolutionary distances estimated from sequences are not additive, but this is a simple and standard approach to compare algorithms and agglomeration criteria.

We use for the correct tree T the phylogeny of 75 placental mammals from [6]. The percentage of missing entries is $P_{\text{miss}} = 1\%, 5\%, 10\%, 20\%, 30\%$. For each P_{miss} value, 500 replicates are randomly generated. From each of these 5×500 incomplete

additive distance matrices, a tree \hat{T} is inferred by FITCH, MW* and BIONJ*. Various values of the s parameter are tested for BIONJ*, in order to compare the topological accuracy of Q_{xy}^* , N_{xy}^* , and of the combination of these two agglomeration criteria. With $s = 1$, BIONJ* uses Q_{xy}^* only. With $s > 1$, the taxon pairs corresponding to the s highest values of Q_{xy}^* are reanalyzed with N_{xy}^* (and with other criteria when ties occur; see Methods). When s becomes large (which is denoted as $s = \max$) BIONJ* uses N_{xy}^* only, as all taxon pairs are retained in the first selection step.

Each inferred tree \hat{T} is compared to the correct tree T by using the quartet distance d_q [47]. This topological distance measures the number of resolved 4-taxon subtrees which are induced by one tree but not the other, and thus is more precise than the widely used bipartition distance [48] which counts the number of internal branches present in one tree but not in the other. Moreover, the quartet distance is less affected than the bipartition distance by small topological errors, e.g. wrong position of a single taxon [49]. This distance is normalized: $d_q = 0$ indicates that T and \hat{T} are identical, whereas $d_q = 1$ means that both trees do not share any resolved 4-taxon subtrees. Averages of the 500 d_q measures for each P_{miss} value are displayed in Figure 1, for FITCH, MW*, and BIONJ* with various s values.

All curves in Figure 1 are decreasing; as expected, the correct tree T is better recovered (i.e. the mean d_q value between \hat{T} and T decreases) as the proportion of missing distance P_{miss} becomes closer to 0. Using N_{xy}^* in BIONJ* greatly improves the agglomeration step; e.g. with $P_{\text{miss}} = 10\%$, mean d_q values of BIONJ* are ~ 0.0015 and ~ 0.0008 , with $s = 1$ and $s = 15$, respectively. However, there is no significant difference between $s = 15$ and $s = \max$ (as assessed by a sign-test [50] based on the 500 replicates, all p -values are much larger than 0.05), meaning that a small value of s (e.g. $s = 15$) seems to be enough to focus on the most relevant pairs, while avoiding the computational burden of using N_{xy}^* only. Further experiments (see below) confirm this finding. FITCH and BIONJ* (with $s = 15$ and $s = \max$) have similar accuracy, while MW* tends to perform better than the other algorithms with these data. However, we shall see that algorithm ordering is different with more realistic simulations. These experiments thus confirm the advantage of combining Q_{xy}^* and

N_{xy}^* within BIONJ*, and similar results (not shown) are obtained with NJ* and MVR*.

Comparison of reconstruction algorithms with distance supermatrices

We re-use a simulation protocol that we have used previously to compare a number of tree-reconstruction methods in a phylogenomic context [6]. This protocol involves generating sequences and evolving them along trees, and is more realistic than the comparison described above. We first summarize this protocol, and then report the results that are obtained with the simulated datasets by FITCH, MW*, NJ*, BIONJ* and MVR*. To estimate the distance supermatrix that is the input of these algorithms, we use the SDM method ([6], see also Methods) which computes a supermatrix that summarizes the topological signal being contained in a collection $\{(\Delta_{ij}^1), (\Delta_{ij}^2), \dots, (\Delta_{ij}^k)\}$ of k distance matrices. Simulations [6] have shown the high-quality of this distance supermatrix in both medium- and high-level gene combinations.

Simulations are as follows (see [6] for more details). Starting from a randomly generated tree T with $n = 48$ taxa, evolution of k genes is simulated, with $k = 2, 4, \dots, 20$. For each of the k genes, some taxa are randomly deleted. Two deletion probabilities are used: 25% to preserve high overlap between the different taxon sets, and 75% to induce low overlap. From these k partially deleted gene alignments, k distance matrices are estimated to compose the collection C_Δ of source matrices. The SDM method is then run with C_Δ to obtain a distance supermatrix corresponding to a medium-level combination of the k partially deleted genes. To study the high-level combination, a phylogenetic tree is inferred by PhyML [17] from each of the k partially deleted genes; then, the path length distance between each taxon pair for each of the k phylogenies is computed, to form the collection C_T of k additive distance matrices that are equivalent to the k PhyML trees. Finally, SDM is applied to C_T to obtain a distance supermatrix corresponding to a high-level gene combination.

This simulation protocol is repeated 500 times for each value of k and each deletion proportion. We obtain this way (10 gene collection sizes \times 500 collections \times

2 overlap conditions \times 2 gene combination levels) = 20,000 distance supermatrices, which are denoted as $(\Delta_{ij}^{\text{SDM}})$ and are frequently incomplete. Indeed, if taxon i is missing for gene p , then Δ_{ij}^p is missing—which is denoted as $\Delta_{ij}^p = \emptyset$ —, and if $\Delta_{ij}^p = \emptyset$ for all $p = 1, 2, \dots, k$, then $\Delta_{ij}^{\text{SDM}} = \emptyset$. With 25% deletion rate, almost all distance supermatrices are complete when $k \geq 14$. With 75% deletion rate, all distance supermatrices are incomplete, but the number of missing distances decreases as k increases (missing distance proportions range from 42% to 11%).

FITCH and MW* are run with default options. In accordance with Figure 1, s is set to 15 for NJ*, BIONJ* and MVR*. With BIONJ*, V_{ij} variances (associated with Δ_{ij}^{SDM} distance estimates) are naturally defined by $V_{ij} \propto \Delta_{ij}^{\text{SDM}}$ if $\Delta_{ij}^{\text{SDM}} \neq \emptyset$, else $V_{ij} = \emptyset$. Variances used by MVR* comply with the same rule, but account for other parameters such as the length and the number of sequences being used to estimate each Δ_{ij}^{SDM} distance (see Methods). Accuracy of the five algorithms is measured with the topological distance d_q , as above, and averaged for the 500 replicates corresponding to each of the conditions. Results are reported in Table 1 for the medium-level gene combination, and in Table 2 for the high-level gene combination. For each value of k , the first- and second-best mean d_q values are indicated in bold&underlined and bold, respectively, and a sign-test [50] based on the 500 replicates is used to assess the significance of the difference between these two best values.

In the medium-level gene combination, NJ* and MW* are outperformed by other algorithms. With a 25% deletion rate, BIONJ* has best topological accuracy, followed by FITCH. However, the sign-test indicates that the difference between these two algorithms is moderately significant as the p -value is lower than 0.05 for only five k values (= 6, 8, 12, 16, and 18). With a 75% deletion rate, FITCH is best, but again the sign-test shows that FITCH, BIONJ* and MVR* are broadly equivalent.

With high-level combination distance supermatrices, NJ* and MW* still tend to be outperformed by other algorithms. BIONJ* is in between, and the best mean d_q values are observed with MVR* which is followed by FITCH. The sign-test

broadly confirms the significance of this observation, though the accuracy difference between MVR* and FITCH is relatively low.

Altogether, these experiments show that MVR* is at least as accurate as FITCH, that BIONJ* has similar performance, while NJ* and MW* are behind these three algorithms. Comparing these findings with the results from [6, Figure 2], we see that (in the high-level framework, Table 2) MVR* is more accurate than the standard Matrix Representation with Parsimony method (MRP, [51,52]), in most cases; e.g. with $k=10$, MVR* has mean d_q values of 0.0171 and 0.0663, for 25% and 75% deletion rate, respectively, while mean d_q values of MRP equal 0.0175 and 0.1152. MVR* (combined with SDM) outperforms MRP with sparse information (75% deletion rate and/or low number of genes), while both approaches are nearly equivalent when the information is abundant (25% deletion rate). An explanation [53] of this finding could be that the distance approach not only uses the topology of the source trees (as MRP) but also their branch lengths. Distance-based supertrees thus contain more information than MRP supertrees, which makes a noticeable difference when the information is sparse, but does not impact much the results with abundant information (see also following simulation results).

Results with simulations based on Driskell et al. [20] dataset

This section aims to measure the accuracy of the different tree building algorithms when applied to simulated datasets being more realistic than those commonly used in a phylogenomic perspective. Most notably, uniformly random gene deletion (used in previous section, following [54]) is not fully realistic because some genes (e.g. cytochrome b) are sequenced for most species, while some other genes are rarely sequenced (or rare among living species). It follows that the gene presence/absence pattern is different with real datasets to this being induced by uniformly random gene deletion (see [20,55-57] for illustrative examples). To this purpose, we use the character supermatrix from Driskell et al. [20], which comprises 69 green plant species and 254 genes, and was built via an automated exploration process of GenBank. This matrix contains a total number of 2777 sequences and has 87% missing characters, which are unequally distributed among taxa. Only 3 taxa have

more than 50% genes, whereas 42 have 10% genes or less. In the same way, a few genes are present in most taxa (e.g., the 2 most sequenced genes belong to 59 taxa), whereas other genes are rare (e.g. 121 genes are present in at most 5 taxa). However, these $k = 254$ genes are complementary and the SDM distance supermatrix only contains $\sim 1.2\%$ missing entries. This low proportion of missing entries is favorable to tree reconstruction, but still requires an algorithm able to deal with incomplete matrices.

We use a simulation protocol analogous to that described above (see also [6]). The only difference is the deletion procedure, with random deletion replaced by the gene presence/absence pattern of [20, Figure 2.B]. We generate 100 datasets this way with $n = 69$ taxa and $k = 254$ genes. From these 100 datasets, we infer 100 distance matrix collections C_Δ and 100 tree collections C_T . Each of these 2×100 collections is dealt with by SDM, to obtain a distance supermatrix $(\Delta_{ij}^{\text{SDM}})$ that contains the same missing entries as those induced by the original dataset [20]. We use these matrices to compare FITCH, MW*, NJ*, BIONJ* and MVR*, based on d_q quartet distance between the correct and inferred trees (see above). Our three algorithms are run with both $s = 15$ and $s = \max$. Results of MRP are also computed, using TNT [58] to infer the most parsimonious trees. TNT is run with 25 random addition sequences, TBR branch swapping and ratchet. The MRP supertree is defined in the standard way [59] as the strict consensus of the most parsimonious trees. Results are displayed in Table 3, which is similar to Tables 1 and 2; the first- and second-best mean d_q values are indicated in bold&underlined and bold, respectively, and sign-tests are used to assess the significance of the differences between MVR* (our best algorithm), FITCH and MRP.

NJ*, BIONJ* and MVR* do not show any significant difference when used with $s = 15$ and $s = \max$ (as assessed by the sign-test, all p -values are much larger than 0.05, results not shown). This confirms the results of the previous experiments to compare our various agglomeration criteria. NJ* has the worst accuracy, especially in the high-level combination framework. MW*, FITCH and BIONJ* show similar performance, while MVR* is best among distance approaches in the two gene combination levels. Moreover, the difference between MVR* and FITCH is highly

significant (sign-test p -value ≈ 0.0). In the high-level framework, MVR* tends to be better than MRP, although the information is quite abundant (254 genes, $\sim 1.2\%$ of missing distances); however, the difference is not significant with 100 replicates (sign-test p -value ≈ 0.2). The results among distance methods are explained by the fact that MVR* uses fairly accurate estimates (V_{ij}^{SDM}) of the variances of the distances in (Δ_{ij}^{SDM}). Indeed, dataset [20] induces a highly heterogeneous distribution of missing sequences, meaning that some distances are well estimated thanks to a large number of sequences, while some others are poorly estimated via a few sequences. This is accounted for by MVR* in (V_{ij}^{SDM}) calculations (see Methods), while MW*, FITCH and BIONJ* lack this information and use inaccurate estimations of (V_{ij}^{SDM}). The difference between these two approaches (i.e. MVR* on the one hand, and MW*, FITCH and BIONJ* on the other hand) is somewhat hidden when using uniformly random sequence deletion, because with the latter all distances are broadly estimated with the same number of genes. With biologically realistic pattern of gene presence/absence, the difference becomes important, especially for the high-level combination. Thus, this last set of simulations confirms the findings of the previous ones and supports the capacity of MVR* for dealing with phylogenomic data.

Run time comparison

Run times with various dataset sizes have been measured on a PC Pentium IV 1.8GHz (1Gb RAM) and are displayed in Table 4. We do not report the runtimes of NJ* and BIONJ*, as they are nearly the same as those of MVR*. In fact, NJ* and BIONJ* are $\sim 2\%$ faster than MVR*, because they are simpler, but these simplifications does not concern the heavy $O(n^3)$ parts of the algorithms (see Methods). We also report the run times of SDM [6], which are in the same range as the fastest tree building algorithms, except with Driskell et al. [20]-like datasets, where SDM has to summarize a large number (254) of source matrices, but where the number of taxa (69) is relatively low. In this case, the run time of SDM is analogous to that of FITCH and MW* and remains quite handy (~ 5 minutes per dataset).

As expected from their mere principle, the run times of the various tree building algorithms are not much affected by the proportion of missing distances,

which is induced by the taxon deletion rate (25% or 75%) and the number of source matrices (k). The only apparent exceptions correspond to $k = 2$ and 75% deletion rate, where all algorithms seem to be quite fast; but in this case the distance supermatrices are of low size (~ 20 , ~ 42 and ~ 85 for n equal to 48, 96 and 192, respectively), which explains this finding. Indeed, in this case it occurs frequently that some taxa have no gene (among 2) in common with any of the other taxa, and such taxa cannot be analyzed as all their distances to the other taxa are missing.

With 25% taxon deletion proportion, $n = 48$ and $k = 10$, run times of ~ 3 hours and ~ 5 hours are required by FITCH and MW*, respectively, to build the 500 trees corresponding to all gene collections in any given gene combination level. The same task, which induces calculations similar to bootstrapping, is achieved in ~ 30 seconds by any of our agglomerative algorithms. The difference between the agglomerative algorithms and the others increases when the number of taxa increases, as expected given that their time complexity are $O(sn^3)$ (i.e. $O(n^3)$ as s is kept constant) and $O(n^4)$ or more, respectively. With 192 taxa, FITCH and MW* require more than 3 hours to build a single tree, while the agglomerative algorithms require less than 1 minute; this run time makes easy to perform a bootstrap study with our algorithms, but pretty much impossible with FITCH or MW*. With even larger datasets (say, above 500 taxa) neither FITCH nor MW* can be used to build a single tree, while our algorithms still run in a few minutes.

Conclusion

Thanks to the ever increasing flow of sequence data, phylogenomic analyses and supertree buildings are more and more frequently used to draw the evolutionary tree of living species. Larger and larger datasets are processed, requiring sophisticated approaches and algorithms. In this context, distance-based methods are quite useful, as they are both very fast and fairly accurate. New techniques, such as SDM [6], allow quickly estimating distance supermatrices that summarize the topological signal being contained in a collection of source distance matrices or gene trees. However, these supermatrices may be incomplete due to low taxon coverage in the selected genes. In

this (common) case, fast distance-based tree building algorithms such as NJ, BIONJ, FASTME or STC are no longer applicable.

This paper presents an adaptation to incomplete distance matrices of several agglomerative algorithms, namely NJ, BIONJ and MVR. We show that the formulae forming the basis of these algorithms can be rewritten to account for missing distances. Moreover, the same holds for the quartet-based pair selection criterion of ADDTREE. Combining both NJ and ADDTREE generalized pair selection criteria, we obtain fast and accurate algorithms that require $O(n^3)$ run times, where n is the number of taxa, i.e. run times that are similar to NJ's. These three novel algorithms, named NJ*, BIONJ* and MVR*, show (in our simulations) topological accuracy similar or higher to that of FITCH and MW*, which are much more time consuming. MVR* appears to be best, followed by BIONJ*. In a phylogenomic context, MVR* accounts for (and benefits from, regarding other algorithms) the fact that gene distribution among species is very heterogeneous, which implies that some distances are accurately estimated (using numerous genes) while some others are poorly estimated (with few genes). Combined with the SDM method [6] to estimate distance supermatrices, MVR* and BIONJ* are relevant alternatives to standard supertree techniques [7], as MRP [51,52]. JAVA implementations of these algorithms are available in PhyD* software and downloadable from [8]. All our datasets are also available from this URL.

Several research directions would deserve to be explored. The variances and covariances of the distance estimates in the distance supermatrix could be accounted for in a more complete and accurate way, e.g. in the line of WEIGHBOR [40] for the pair selection criterion, or using the generalized least-squares version of MVR [4]. There is a clear need for a pair selection criterion being able to point out xy taxon pairs, even when the corresponding Δ_{xy} distance is missing. Theoretical results highlighting the cases where our algorithms will succeed (or fail) in recovering the correct tree, would likely help to improve these algorithms or design new ones. Adapting to missing distances very fast algorithms (e.g. [41-46]) could be promising. Finally, dealing with missing distances is likely required in other (non phylogenomic)

applications of phylogenetic trees, and in related problems, as phylogenetic network inference (e.g. [60]).

Methods

Existing agglomerative algorithms are defined by criteria and formulae which all can be rewritten as distance averages. These algorithms (e.g. NJ [1,2], BIONJ [3] and MVR [4]) are generalized to incomplete distance matrices by estimating these averages using available distances, when some of those are missing. In the following, we first define notation and present a generic agglomerative scheme that covers all the algorithms being discussed here. Then, we describe for each of the three agglomeration steps (pair selection, branch length estimation, and matrix reduction), how NJ is generalized into NJ* to deal with missing distances. NJ* is further refined by BIONJ* that incorporates a first simple estimation of the variance associated to each evolutionary distance. Finally, a second, more accurate estimation of this variance is used by MVR* that generalizes the weighted least-squares (WLS) version of the MVR [4] approach.

Notation

Let $L_n = \{1, 2, \dots, n\}$ be the set of all taxa numbered from 1 to n , and (Δ_{ij}) a distance matrix, where Δ_{ij} corresponds to the evolutionary distance between taxa $i, j \in L_n$, and $\Delta_{ii} = 0, \forall i \in L_n$. Distance-based algorithms build a tree T (also denoted as \hat{T} , depending on the context) from (Δ_{ij}) , and estimate all branch lengths T_{uv} , where uv is any pair of sibling nodes in T . At each agglomeration stage, a taxon pair xy is selected, connected to a new internal node u , and replaced by u in (Δ_{ij}) . Thus, at each stage, the set $L_r = \{1, 2, \dots, r\}$ of non-agglomerated taxa drops in cardinality by 1, and r is changed into $r - 1$. Tree reconstruction stops when $r = 2$.

Agglomerative algorithms with complete distance matrices

A number of existing agglomerative algorithms to deal with complete matrices can be summarized using the following scheme [4]:

- Input $L_n = \{1, 2, \dots, n\}$ and (Δ_{ij}) ;
- $r = n$;
- While $r > 2$, do:
 - (a) Select the xy pair to be merged into u by optimizing an agglomeration criterion;
 - (b) Estimate the branch lengths T_{xu} and T_{yu} :

$$T_{xu} = \Delta_{xy} - T_{yu} = \frac{1}{2}\Delta_{xy} + \sum_{i \in L_r - \{x, y\}} w_i (\Delta_{xi} - \Delta_{yi}) \quad (1)$$
 with $\sum_{i \in L_r - \{x, y\}} w_i = \frac{1}{2}$;
 - (c) Reduce the distance matrix (Δ_{ij}) for all $i \neq x, y$:

$$\Delta_{ui} = \lambda_i (\Delta_{xi} - T_{xu}) + (1 - \lambda_i) (\Delta_{yi} - T_{yu}) \quad (2)$$
 with $\lambda_i \in [0, 1]$;
 - (d) $r = r - 1$;
- Output T .

Step (a) in this generic scheme searches for the taxon pair xy to be merged by optimizing an agglomeration criterion. NJ, BIONJ and MVR select the pair which maximizes [1,2]:

$$Q_{xy} = R_x + R_y - (r - 2)\Delta_{xy}, \quad \text{where} \quad R_z = \sum_{i \in L_r} \Delta_{zi}. \quad (3)$$

Let (Δ_{ij}) be additive [61], i.e. be defined as the path-length distance between taxa in a phylogenetic tree T with positive branch lengths; then, maximizing Q_{xy} over all taxon pairs selects a cherry of T , i.e. a pair of taxa being separated by a unique internal node in T . In other words, Q_{xy} is consistent (e.g. [36]). However, it is easily shown (using counter-examples) that the second best taxon pair (based on Q_{xy} values) is not necessarily a cherry of T .

Conversely, the ADDTREE [5] pair selection criterion implies that all cherries of T have highest criterion value. The ADDTREE criterion counts the number of times where the xy pair is a cherry in all taxon quartets $xyij$:

$$N_{xy} = \sum_{i < j \in L_r - \{x, y\}} H(\Delta_{xi} + \Delta_{yj} - \Delta_{xy} - \Delta_{ij}) H(\Delta_{xj} + \Delta_{yi} - \Delta_{xy} - \Delta_{ij}) \quad (4)$$

where $H(t) = 1$ if $t \geq 0$, and $H(t) = 0$ if $t < 0$. This criterion has integer values ranging from 0 to $(n-2)(n-3)/2$, and this maximum value is reached for all cherries (but for the cherries only) with additive distance matrices. Careful implementation [39] of ADDTREE allows for $O(n^4)$ run time. NJ, BIONJ and MVR are much faster. They first compute all R_z sums in Equation (3), and then compute in $O(1)$ the Q_{xy} value of each xy pair. Each agglomeration stage thus requires $O(r^2)$ time (branch-length estimation and matrix reduction are achieved in $O(r)$), and the whole algorithm is in $O(n^3)$. Moreover, Q_{xy} can be seen as a continuous version of N_{xy} [62].

After xy pair selection, x and y are connected to the new node u , and the lengths of xu and yu branches are estimated using Equation (1). Assuming that (Δ_{ij}) is additive and corresponds to tree T , we have $T_{xu} = (\Delta_{xy} + \Delta_{xi} - \Delta_{yi})/2$, $\forall i \neq x, y$. Equation (1) averages these elementary estimators using various (w_i) weightings. With NJ, the average is equally-weighted and we have $w_i = w = 1/(2(r-2))$. We shall see that MVR uses different w_i weights.

Finally (step (c)), (Δ_{ij}) is reduced by replacing x and y with the new node u , and by computing all Δ_{ui} distances, $\forall i \neq x, y$. When (Δ_{ij}) is additive and corresponds to tree T , we have $\Delta_{ui} = \Delta_{xi} - T_{xu} = \Delta_{yi} - T_{yu}$. Equation (2) averages these two elementary estimators. NJ uses equal weights ($\lambda_i = 1 - \lambda_i = 1/2$) while BIONJ and MVR adjust λ_i in order to minimize the variance of Δ_{ui} and to have reliable distance estimates during all agglomeration stages. For this purpose, BIONJ and MVR use (approximate) models for the variances and covariances of the distance estimates in (Δ_{ij}) .

NJ*: generalizing NJ to incomplete distance matrices

When (Δ_{ij}) is incomplete (missing entries are denoted as \emptyset), the criteria and equations above do not apply. We shall see in this section how they are generalized to define the NJ* algorithm, which keeps NJ's $O(n^3)$ time complexity and is nearly equivalent to NJ with complete matrices.

(a) Agglomeration criterion

Let $Q'_{xy} = Q_{xy}/(r-2)$. Maximizing Q'_{xy} is the same as maximizing Q_{xy} (Equation (3)), and we have:

$$Q'_{xy} = \frac{R_{xy}}{r-2} - \Delta_{xy}, \quad \text{where} \quad R_{xy} = \sum_{i \in L_r} (\Delta_{xi} + \Delta_{yi}),$$

which can be rewritten as:

$$Q'_{xy} = \frac{2}{r-2} \Delta_{xy} + \frac{1}{r-2} \sum_{i \in L_r - \{x,y\}} (\Delta_{xi} + \Delta_{yi} - \Delta_{xy}). \quad (5)$$

The sum in Equation (5) relates to terms representing how distant is the path joining x to y from other taxa $i \neq x, y$ ($\Delta_{xi} + \Delta_{yi} - \Delta_{xy}$ equals twice the distance between u and i), whereas the first term expresses the additional distance induced by Δ_{xy} . It has been shown [63,64] that the relative weight of these two factors is unique, due to consistency requirement, and Q'_{xy} can be interpreted as the mean acentrality of the xy pair [35,36]. To extend this criterion to incomplete distance matrices, we estimate it using the set of taxa with non-missing distances: $S_{xy}^* = \{i \in L_r : \Delta_{xi}, \Delta_{yi} \neq \emptyset\}$. Moreover, we assume $\Delta_{xy} \neq \emptyset$, and thus $x, y \in S_{xy}^*$. The normalization factor is then equal to $|S_{xy}^*| - 2$ (instead of $r - 2$) and we obtain the following generalization of Equation (5):

$$Q_{xy}^* = \frac{2}{|S_{xy}^*| - 2} \Delta_{xy} + \frac{1}{|S_{xy}^*| - 2} \sum_{i \in S_{xy}^* - \{x,y\}} (\Delta_{xi} + \Delta_{yi} - \Delta_{xy}),$$

which applies to incomplete distance matrices, and is identical to Q'_{xy} with complete ones. This equation further simplifies into:

$$Q_{xy}^* = \frac{R_{xy}^*}{|S_{xy}^*| - 2} - \Delta_{xy}, \quad \text{where} \quad R_{xy}^* = \sum_{i \in S_{xy}^*} (\Delta_{xi} + \Delta_{yi}). \quad (6)$$

Other solutions are possible to extend Equation (5), e.g. preserving $\Delta_{xy}/(r-2)$ term rather than transforming it into $\Delta_{xy}/(|S_{xy}^*|-2)$. Simulation results (not shown) indicate that criterion (6) has better topological accuracy than these alternatives. Theoretical results would be desirable to explain these observations and establish the properties of criterion (6), but a first simple explanation is that Equation (6) precisely corresponds to the Q'_{xy} value being computed on S_{xy}^* taxon subset. To be consistent on the whole set of taxa (L_r), it is mandatory that the criterion is consistent on taxon subsets (S_{xy}^* , here), and Equation (6) satisfies this requirement.

Maximizing Q_{xy}^* seems to require $O(r^3)$ time for each iteration, and thus a total time complexity of $O(n^4)$. However, efficient implementation allows for $O(n^3)$ total run time. At the first stage ($r = n$), R_{xy}^* and $|S_{xy}^*|$ values are computed and stored for all $x, y \in L_n$, which requires $O(n^3)$ time. In the subsequent agglomeration stages, these values are updated as follows:

- After step (a), for all $i, j \in L_r - \{x, y\}$ we remove from R_{ij}^* and $|S_{ij}^*|$: Δ_{xi} and Δ_{xj} (if $\Delta_{xi} \neq \emptyset$ and $\Delta_{xj} \neq \emptyset$), and Δ_{yi} and Δ_{yj} (if $\Delta_{yi} \neq \emptyset$ and $\Delta_{yj} \neq \emptyset$).
- After step (c), we compute R_{ui}^* and $|S_{ui}^*|$ for all $i \in L_r - \{u\}$, and
- for all $i, j \in L_r - \{u\}$, we add Δ_{ui} and Δ_{uj} to R_{ij}^* and $|S_{ij}^*|$ (if $\Delta_{ui} \neq \emptyset$ and $\Delta_{uj} \neq \emptyset$).

Each of these three updating routines requires $O(r^2)$ time, just as pair selection using criterion (6), meaning that using Q_{xy}^* instead of Q_{xy} does not change the total $O(n^3)$ time complexity of the original NJ algorithm.

However, as discussed earlier, a limitation of criterion Q_{xy}^* is that: (1) it cannot be computed when $\Delta_{xy} = \emptyset$, and (2) only the best pair is guaranteed (with additive distance) to be a cherry in the correct tree. When xy is the best pair in the complete additive distance matrix, but Δ_{xy} is missing in the available distance matrix, then using Q_{xy}^* does not provide any guaranty of correctness. This difficulty is partly alleviated when using a generalization of N_{xy} , as this criterion selects all cherries in the correct tree with complete additive distances. When some of the cherries correspond to missing distances, we are still able to select the others that correspond

to non-missing entries. Our generalization of N_{xy} (Equation (4)) to incomplete distances is defined as follows. Let:

$$\tilde{N}_{xy}^* = \sum_{i,j \in C_{xy}^*} H(\Delta_{xi} + \Delta_{yj} - \Delta_{xy} - \Delta_{ij}), \quad (7)$$

$$\text{where } C_{xy}^* = \{(i, j) \neq (x, y), (y, x) : i \neq j, \Delta_{xi}, \Delta_{yj}, \Delta_{ij} \neq \emptyset\}. \quad (8)$$

\tilde{N}_{xy}^* differs from N_{xy} in that we sum both H terms, instead of multiplying them. This way we exploit all available information. Indeed, when $\Delta_{xj} = \emptyset$ and/or $\Delta_{yi} = \emptyset$, but the other entries are available, we still use $H(\Delta_{xi} + \Delta_{yj} - \Delta_{xy} - \Delta_{ij})$ in \tilde{N}_{xy}^* while a multiplicative solution in the line of N_{xy} would discard this term. Moreover, it is easily seen that $\tilde{N}_{xy}^* = 2N_{xy}$ with complete additive distances. To select among taxon pairs, we use the averaged form of \tilde{N}_{xy}^* , that is:

$$N_{xy}^* = \frac{\tilde{N}_{xy}^*}{|C_{xy}^*|}, \quad (9)$$

which expresses the mean number of quartets where the xy pair corresponds to a cherry.

However, selecting pairs using N_{xy}^* sometimes produces ties. In this case, we select the pair with higher $|C_{xy}^*|$ value, that is the pair which is supported by the larger number of quartets. But ties may still occur, in which case we use:

$$M_{xy}^* = |Miss(x) - Miss(y)| + |Miss(y) - Miss(x)|, \quad (10)$$

where $Miss(z) = \{i \in L_r, \neq z : \Delta_{iz} = \emptyset\}$ corresponds to missing entries for taxon z . M_{xy}^* counts the number of missing entries in the current matrix that will be removed in the next step (see reduction procedure (13)). Maximizing M_{xy}^* tends to quickly fill missing entries in the running distance matrix, which both frees from $\Delta_{xy} \neq \emptyset$ limitation and allows using Q_{xy} pair selection criterion only. Finally, in some (very rare) cases, we still have ties and then maximize the continuous version [62] of \tilde{N}_{xy}^* :

$$N_{xy}^{**} = \sum_{(i,j) \in C_{xy}^*} (\Delta_{xi} + \Delta_{yj} - \Delta_{xy} - \Delta_{ij}). \quad (11)$$

Pair selection criteria N_{xy}^* (9), $|C_{xy}^*|$ (8), M_{xy}^* (10) and N_{xy}^{**} (11) are used in a lexicographic way: taxon pairs are ranked based on the first criterion (\tilde{N}_{xy}^*), the second one ($|C_{xy}^*|$) is used in case of ties, etc. However, using these four criteria only would result in $O(n^4)$ time complexity. In order to preserve $O(n^3)$ run times, we first select the s top pairs based on Q_{xy}^* criterion (6), and then use the other criteria in lexicographic order to select the pair to be agglomerated among these s pairs. As computing Equations (7) to (11) requires $O(r^2)$ or less per taxon pair, the total time complexity of pair selection is $O(n^3)$ (1)(first selection using (6)) plus $O(s \sum r^2)$ (final selection using (8) to (11)), i.e. $O(sn^3)$. As explained above, Q_{xy}^* does not provide any guaranty of correctness with missing distances, while N_{xy}^* and N_{xy}^{**} partly circumvent the difficulty. However, Q_{xy}^* enables to extract the most promising pairs for agglomeration and we have seen (Figure 1) that using for s a small constant (typically 15) is sufficient to obtain high accuracy, meaning that, in practice, run times are in $O(n^3)$.

(b) Branch length estimation

Equation (1) is easily rewritten using non-missing entries only:

$$T_{xu} = \Delta_{xy} - T_{yu} = \frac{1}{2} \Delta_{xy} + \sum_{i \in S_{xy}^* - \{x,y\}} w_i (\Delta_{xi} - \Delta_{yi}), \quad \text{where} \quad \sum_{i \in S_{xy}^* - \{x,y\}} w_i = 1/2. \quad (12)$$

NJ uses the same weight w_i for every taxon i . The same holds for NJ*, that is, $w_i = w = 1 / (2(|S_{xy}^*| - 2))$. Note that for the selected pair we have $\Delta_{xy}, S_{xy}^* \neq \emptyset$, meaning that Equation (12) is always applicable. Just as with NJ, branch length estimation (12) requires $O(r)$ time at each agglomeration stage.

(c) Matrix reduction

Equation (2) averages two elementary estimators, and with NJ this average is equally weighted. With missing distances it may occur that one of these two estimators is not applicable (e.g. when $\Delta_{xi} = \emptyset$), that both are applicable, or that none is applicable. Thus, in NJ* Equation (2) becomes:

$$\Delta_{ui} = \begin{cases} \lambda_i (\Delta_{xi} - T_{xu}) + (1 - \lambda_i) (\Delta_{yi} - T_{yu}) & \text{when } \Delta_{xi} \neq \emptyset \text{ and } \Delta_{yi} \neq \emptyset, \\ \Delta_{xi} - T_{xu} & \text{when } \Delta_{xi} \neq \emptyset \text{ and } \Delta_{yi} = \emptyset, \\ \Delta_{yi} - T_{yu} & \text{when } \Delta_{xi} = \emptyset \text{ and } \Delta_{yi} \neq \emptyset, \\ \emptyset & \text{when } \Delta_{xi} = \Delta_{yi} = \emptyset, \end{cases} \quad (13)$$

where $\lambda_i = \lambda = 1/2$. In the second and third cases, entries missing in the previous matrix are now present in the new, reduced matrix. We have seen that criterion (10) tends to maximize the number of such entries, in order to fill as fast as possible the missing distances in the running matrix. Just as branch length estimation (12), matrix reduction (13) requires $O(r)$ time at each stage and does not impact total time complexity. Thus, NJ* requires $O(n^3)$ run times, when s is kept constant.

BIONJ*: improving the reduction step, a first simple solution

BIONJ* uses the same pair selection criteria as NJ*, and adapts to missing distances BIONJ reduction procedure. BIONJ uses the degree of freedom corresponding to the λ_i parameter in Equation (2), in order to minimize the variance of the new Δ_{ui} estimates in step (c). For this purpose, BIONJ assumes a simple Poisson model of the variances in the original (Δ_{ij}) matrix, stating that the variance V_{ij} of Δ_{ij} is proportional to Δ_{ij} . BIONJ also accounts for the covariances in (Δ_{ij}) (see [3] for more details). It uses a single λ parameter for every xy pair, which does not depend on i and is given by

$$\lambda_i = \lambda = \frac{1}{2} + \frac{1}{2(r-2)V_{xy}} \sum_{j \in L_r - \{x,y\}} (V_{yj} - V_{xj}). \quad (14)$$

Again, this equation may be seen as an average and can be rewritten using available entries only as:

$$\lambda_i^* = \lambda^* = \frac{1}{2} + \frac{1}{2(|S_{xy}^*| - 2)V_{xy}} \sum_{j \in S_{xy}^* - \{x,y\}} (V_{yj} - V_{xj}). \quad (15)$$

The reduction step (c) is achieved by BIONJ* as defined by Equation (13), but using so-defined λ^* (instead of $1/2$) when $\Delta_{xi} \neq \emptyset$ and $\Delta_{yi} \neq \emptyset$.

Moreover, BIONJ starts with variance matrix $(V_{ij}) = (\Delta_{ij})$ and reduces this matrix at each stage using λ value from Equation (14) and equation:

$$V_{ui} = \lambda V_{xi} + (1 - \lambda) V_{yi} - \lambda(1 - \lambda) V_{xy}.$$

BIONJ* combines this formula with Equation (13) and (15) to reduce the variance matrix, that is:

$$V_{ui} = \begin{cases} \lambda^* V_{xi} + (1 - \lambda^*) V_{yi} - \lambda^* (1 - \lambda^*) V_{xy} & \text{when } \Delta_{xi} \neq \emptyset \text{ and } \Delta_{yi} \neq \emptyset, \\ V_{xi} & \text{when } \Delta_{xi} \neq \emptyset \text{ and } \Delta_{yi} = \emptyset, \\ V_{yi} & \text{when } \Delta_{xi} = \emptyset \text{ and } \Delta_{yi} \neq \emptyset, \\ \emptyset & \text{when } \Delta_{xi} = \Delta_{yi} = \emptyset. \end{cases} \quad (16)$$

Computing λ^* using Equation (15) and achieving matrix reductions (13) and (16) requires $O(r)$ run times. Thus, BIONJ* has $O(n^3)$ time complexity (when s is kept constant, else $O(sn^3)$).

MVR*: improving BIONJ* using variances dedicated to distance supermatrices

The BIONJ variance model is well suited for one-gene studies where distance estimations all use the same number of sites (at least when gaps are removed). With phylogenomic studies, some distances are computed using a large number of genes, and thus are reliable, while other distances are based on a few genes and are poorly estimated. Moreover, some distances may be missing due to the absence of common genes between the two species being compared. Altogether, this implies that the BIONJ and BIONJ* variance model can be improved to better fit phylogenomic requirements. This section describes the MVR* algorithm that is intended to this purpose.

Steps (b) and (c) in the generic scheme are based on w_i and λ_i parameters, respectively. The MVR algorithm [4] generalizes the BIONJ approach and uses these degrees of freedom in order to minimize the variance of the new estimates T_{ux} , T_{uy} and Δ_{ui} . The main difference from BIONJ is that MVR is able to deal with any variance-covariance model of the Δ_{ij} distance estimates, while BIONJ is restricted to the Poisson model. The MVR variant that we use here only considers the variances

and neglects the covariances, thus assuming a weighted least-squares model (it was called MVR-WLS in [4], but is named MVR here for simplicity). Thus, MVR inputs a distance matrix (Δ_{ij}) and the corresponding (V_{ij}) variance matrix. We shall see in the next section how (V_{ij}) is calculated to deal with phylogenomic data, and describe now the way MVR and MVR* use and update these matrices all along the agglomeration procedure.

MVR uses Q_{xy} pair selection criterion (3), just as NJ and BIONJ, while MVR* uses the same criteria and selection procedure as NJ* and BIONJ*.

In MVR step (b), i.e. branch length estimation, w_i weights in Equation (1) depend on i and are given by:

$$w_i = \frac{\mu}{V_{xi} + V_{yi}},$$

$$\text{with normalization term } \mu = \frac{1}{2} \left(\sum_{i \in L_r - \{x,y\}} \frac{1}{V_{xi} + V_{yi}} \right)^{-1}. \quad (17)$$

MVR* uses Equation (12) (instead of Equation (1)) to deal with missing entries, and adapts above Equation (17) by replacing L_r by S_{xy}^* .

In MVR step (c), i.e. matrix reduction, a different λ_i parameter is associated in Equation (2) to each taxon $i \neq x, y$ using:

$$\lambda_i = \frac{V_{yi}}{V_{xi} + V_{yi}}. \quad (18)$$

This value puts more weight and confidence on $(\Delta_{xi} - T_{xu})$ when the associated variance V_{xi} is low, compared to V_{yi} . Equation (18) is also used by MVR*, but combined with Equation (13) to deal with missing distances.

Finally, MVR (just like BIONJ) reduces the variance matrix at each agglomeration stage. To this purpose, MVR uses the following equation:

$$V_{ui} = \frac{V_{xi} V_{yi}}{V_{xi} + V_{yi}}.$$

This equation is also used by MVR* in combination with Equation (16).

All the computations described above (except pair selection) require $O(r)$ run times at each agglomeration stage, and thus MVR* has $O(n^3)$ time complexity, just as do NJ* and BIONJ*.

Estimating the variances associated to distance supermatrices

Distance supermatrices are computed [6,19] from source matrices which are first rescaled, and then averaged. SDM [6] inputs a collection $C = \{(\Delta_{ij}^1), (\Delta_{ij}^2), \dots, (\Delta_{ij}^k)\}$ of k distance matrices —each defined on taxon set L_p and estimated from sequences with size s_p —, and deforms them, without changing their topological signal, so as to bring them as close as possible to each other before averaging. The first deformation is scaling, which multiplies each (Δ_{ij}^p) distance matrix by a factor α_p . The second (optional in SDM) deformation adds a constant a_p to every non-diagonal Δ_{ij}^p entries. Then, SDM averages the resulting modified matrices to obtain the $(\Delta_{ij}^{\text{SDM}})$ supermatrix that is defined by:

$$\Delta_{ij}^{\text{SDM}} = \frac{1}{W_{ij}} \sum_{1 \leq p \leq k, L_p \ni \{i,j\}} s_p (\alpha_p \Delta_{ij}^p + a_{ip} + a_{jp}), \text{ where } W_{ij} = \sum_{1 \leq p \leq k, L_p \ni \{i,j\}} s_p. \quad (19)$$

Neglecting the variance of the deformation factors, we obtain a simple expression of the variance of Δ_{ij}^{SDM} :

$$V_{ij}^{\text{SDM}} = \frac{1}{W_{ij}^2} \sum_{1 \leq p \leq k, L_p \ni \{i,j\}} s_p^2 \alpha_p^2 V_{ij}^p, \quad (20)$$

where V_{ij}^p is the variance of Δ_{ij}^p . Note that no covariance terms between any Δ_{ij}^p and Δ_{ij}^q estimates appear in Equation (20), as these source distances are estimated from different genes and are independent. Moreover, the covariances between the entries in the SDM supermatrix are neglected, as is the case in a number of (WLS) approaches, e.g. [30,32,40].

Several studies have shown that the variance V_{ij} associated with the evolutionary distance Δ_{ij} (estimated from a single gene) is approximately equal to Δ_{ij}^ρ / s with $\rho \approx 2$ [11,65]. Based on various experiments (not shown), we have

chosen the usual formula $V_{ij} = \Delta_{ij}^2 / s$, which corresponds to default option in FITCH program. Equation (20) then becomes:

$$V_{ij}^{\text{SDM}} = \frac{1}{W_{ij}^2} \sum_{1 \leq p \leq k, L_p \ni \{i,j\}} s_p (\alpha_p \Delta_{ij}^p)^2.$$

Authors' contributions

AC designed and implemented the algorithms and experiments, performed the computations that are shown here, and wrote the manuscript. OG supervised these works, participated in the design of algorithms and experiments, and wrote the manuscript.

Acknowledgements

Sincere thanks to Vincent Berry, Richard Desper, Emmanuel J.P. Douzery and two anonymous referees for their suggestions and comments. This research was supported by SUPERTREE project of ACI-IMPBIO. Part of the work was carried out when OG participated to the Phylogenetics programme at Isaac Newton Institute for Mathematical Sciences, Cambridge, UK.

References

1. Saitou N, Nei M: **The neighbor-joining method: a new method for reconstructing phylogenetic trees.** *Mol. Biol. Evol.* 1987, 4: 406-425
2. Studier JA, Kepler KJ: **A note on the neighbor-joining method of Saitou and Nei.** *Mol. Biol. Evol.* 1988, 5: 729-731
3. Gascuel O: **BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data.** *Mol. Biol. Evol.* 1997, 14: 685-695
4. Gascuel O: **Data model and classification by trees: the minimum variance reduction (MVR) method.** *J. Classif.* 2000, 17(1): 67-99

5. Sattath S, Tversky A: **Additive similarity trees.** *Psychometrika* 1977, 42: 319-345
6. Criscuolo A, Berry V, Douzery EJP, Gascuel O: **SDM: a fast distance-based approach for (super)tree building in phylogenomics.** *Syst. Biol.* 2006, 55(5): 740-755
7. Bininda-Emonds ORP: *Phylogenetic supertree: combining information to reveal the tree of life.* Kluwer Academic, New York, 2004
8. **Downloadable PhyD* software from ATGC Montpellier bioinformatics platform** [<http://www.atgc-montpellier.fr/phyd/>]
9. Sourdis J, Nei M: **Relative efficiencies of the maximum parsimony and distance-matrix methods in obtaining the correct phylogenetic tree.** *Mol. Biol. Evol.* 1988, 5(3): 298-311
10. Saitou N, Imanishi T: **Relative efficiencies of the Fitch-Margoliash, maximum-parsimony, maximum-likelihood, minimum-evolution and neighbor-joining methods of phylogenetic tree construction in obtaining the correct tree.** *Mol. Biol. Evol.* 1989, 6(5): 514-525
11. Kuhner MK, Felsenstein J: **A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates.** *Mol. Biol. Evol.* 1994, 11: 459-468
12. Strimmer K, von Haeseler A: **Accuracy of neighbor-joining for n-taxon trees.** *Syst. Biol.* 1996, 45(4): 516-523
13. Nei M, Kumar S, Takahashi K: **The optimization principle in phylogenetic analysis tends to give incorrect topologies when the number of nucleotides or amino acids used is small.** *Proc. Nat. Ac. Sc.* 1998, 95: 12390-12397

14. Kumar S, Gadagkar SR: **Efficiency of the neighbor-joining method in reconstructing deep and shallow evolutionary relationship in large phylogenies.** *J. Mol. Evol.* 2000, 51(6): 544-553
15. Takahashi K, Nei M: **Efficiencies of fast algorithms of phylogenetic inference under the criteria of maximum parsimony, minimum evolution, and maximum likelihood when a large number of sequences are used.** *Mol. Biol. Evol.* 2000, 17: 1251-1258
16. Rosenberg MS, Kumar S: **Traditional phylogenetic reconstruction methods reconstruct shallow and deep evolutionary relationship equally well.** *Mol. Biol. Evol.* 2001, 18(9): 1823-1827
17. Guindon S, Gascuel O: **A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood.** *Syst. Biol.* 2003, 52(5): 696-704
18. Swofford DL, Olsen GJ, Waddell PJ, Hillis DM: **Phylogenetic inference.** In: *Molecular Systematics.* Edited by Hillis DM, Moritz C, Mable BK. Sinauer Associates, Massachusetts, 1996, 407-509
19. Lapointe F-J, Cucumel G: **The average consensus procedure : combination of weighted trees containing identical or overlapping sets of taxa.** *Syst. Biol.* 1997, 46(2): 306-312
20. Driskell AC, Ané C, Burleigh JG, McMahon MM, O'Meara BC, Sanderson MJ: **Prospects for building the tree of life from large sequence databases.** *Science* 2004, 306: 1172-1174
21. Philippe H, Snell EA, Baptiste E, Lopez P, Holland PWH, Casane D: **Phylogenomics of eukaryotes: Impact of missing data on large alignments.** *Mol. Biol. Evol.* 2004, 21(9):1740-1752

22. Galtier N: **A model of horizontal gene transfer and the bacterial phylogeny problem.** *Syst. Biol.* 2007, 56(4):633-642
23. Steel M, Rodrigo A: **Maximum-likelihood Supertrees.** *Syst. Biol.* 2008, in press
24. Farach M, Kannan S, Warnow T: **A robust model for finding optimal evolutionary trees.** *Algorithmica* 1995, 13: 155-179
25. De Soete G: **Ultrametric tree representations of incomplete dissimilarity data.** *J. Clasif.* 1984, 1: 235-242
26. Landry P-A, Lapointe F-J, Kirsch JAW: **Estimating phylogenies from lacunose distance matrices: additive is superior to ultrametric estimation.** *Mol. Biol. Evol.* 1996, 13: 818-823
27. Lapointe F-J, Landry P-A: **A fast procedure for estimating missing distances in incomplete matrices prior to phylogenetic analysis.** In *Currents computational molecular biology*, edited by El-Mabrouk N, Lengauer T, Sankoff D. Publications CRM Montréal, 2001, 189-190.
28. Guénoche A, Grandcolas S: **Approximations par arbre d'une distance partielle.** *Math. Inf. Sc. Hum.* 1999, 146:51-64
29. Makarenkov V. **TREX: Reconstructing and visualizing phylogenetic trees and reticulation networks.** *Bioinformatics* 2001, 17:664–668
30. Felsenstein F: **An alternating least-squares approach to inferring phylogenies.** *Syst. Biol.* 1997, 46: 101-111
31. Felsenstein J. **PHYLIP: Phylogeny inference package, version 3.6b.** Distributed by the author. University of Washington, Seattle, 1993

32. Makarenkov V, Leclerc B. **An algorithm for the fitting of a phylogenetic tree according to a weighted least-squares criterion.** *J. Classif.* 1999, 16:3–26
33. Makarenkov V, Lapointe F-J: **A weighted least-squares approach for inferring phylogenies from incomplete distance matrices.** *Bioinformatics* 2004, 20: 2113-2121
34. Gascuel O, Steel M: **Neighbor Joining Revealed.** *Mol. Biol. Evol.* 2006, 23(11): 1997-2000
35. Mirkin B. *Mathematical classification and clustering.* Kluwer Academic, London, 1996
36. Gascuel O: **Concerning the NJ algorithm and its unweighted version, UNJ.** In: *Mathematical Hierarchies and Biology.* Edited by Mirkin B, McMorris FR, Roberts FS, Rzhetsky A. DIMACS series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Providence, 1997, 140-170
37. Zaretskii K: **Postroenie dereva po naboru rasstoianii mezhdru visiacimi vershinami** (Constructing a tree on the basis of a set of distances between the hanging vertices; in Russian). *Uspehi Matematicheskikh Nauk* 1965, 20: 90-92
38. Buneman P: **The recovery of trees from measures of dissimilarity.** In: *Mathematics in archaeological and historical sciences.* Edited by Hudson F, Kendall D, Tautu P. Edinburgh University Press 1971, 387-395
39. Elemento O, Gascuel O: **A fast and accurate distance algorithm to reconstruct tandem duplication trees.** *Bioinformatics* 2002, 18: 92-99

40. Bruno WJ, Succi ND, Halpern AL: **Weighted neighbor joining: a likelihood-based approach to distance-based phylogeny reconstruction.** *Mol. Biol. Evol.* 2000, 17(1): 189-197
41. Howe K, Bateman A, Durbin R: **QuickTree: building huge Neighbor-Joining trees of protein sequences.** *Bioinformatics.* 2002, 18: 1546-1547
42. Desper R, Gascuel O: **Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle.** *J. Comp. Biol.* 2002, 19(5): 687-705
43. Mailund T, Pedersen CN: **QuickJoin—fast neighbour-joining tree reconstruction.** *Bioinformatics.* 2004, 20: 3261-3262
44. Vinh le S, von Haeseler A: **Shortest triplet clustering: reconstructing large phylogenies using representative sets.** *BMC Bioinf.* 2005, 8: 92
45. Elias I, Lagergren J: **Fast neighbor joining.** In: *Proceedings of ICALP 2005.* Edited by Caires L, Italiano GF, Monteiro L, Palamidessi C, Yung M. LNCS 3580, Springer, Lisbon, 2005, 1263-1274
46. Evans J, Sheneman L, Foster JA: **Relaxed neighbor joining: a fast distance-based phylogenetic tree construction method.** *J. Mol. Evol.* 2006, 62: 785-792
47. Estabrook GF, McMorris FR, Meacham CA: **Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units.** *Syst. Zool.* 1985, 34: 193-200
48. Robinson D, Foulds L: **Comparison of weighted labeled trees.** *Lect. Notes Math.* 1979, 748: 119–126
49. Steel MA, Penny D: **Distribution of tree comparison metrics—Some new results.** *Syst. Biol.* 1993, 42: 126–141

50. Dixon WJ, Mood AM: **The statistical sign test.** *J. Am. Statist. Assoc.* 1946, 41: 557-566
51. Baum BR: **Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees.** *Taxon* 1992, 41:3–10
52. Ragan MA: **Phylogenetic inference based on matrix representation of trees.** *Mol. Phylogenet. Evol.* 1992, 1:53–58
53. Lapointe FJ, Cucumel G: **The average consensus procedure: combination of weighted trees containing identical or overlapping sets of taxa.** *Syst. Biol.* 1997, 46:306-312
54. Eulenstein O, Chen D, Burleigh JD, Fernandez-Baca D, Sanderson HJ: **Performance of flip supertree construction with a heuristic algorithm.** *Syst. Biol.* 2004, 53:299–308
55. Gatesy J, Matthee C, DeSalle R, Hayashi C: **Resolution of a supertree/ supermatrix paradox.** *Syst. Biol.* 2002, 51:652–664
56. Beck RMD, Bininda-Emonds ORP, Cardillo M, Liu FR, Purvis A: **A higher level supertree of placental mammals.** *BMC Evol. Biol.* 2006, 6: 93
57. MacMahon MM, Sanderson MJ: **Phylogenetic supermatrix analysis of GenBank sequences from 2228 Papilionid legumes.** *Syst. Biol.* 2006, 55(5): 818-836
58. Goloboff P, Farris J, Nixon K: **TNT: Tree analysis using new technology,** distributed by the authors, 2003.
59. Bininda-Emonds ORP, Bryant NH: **Properties of matrix representation with parsimony analyses.** *Syst. Biol.* 1998, 47:497-508

60. Chan H-L, Jansson J, Lam T-W, Yiu S-M. **Reconstructing an ultrametric galled phylogenetic network from a distance matrix.** In: *Mathematical Foundation of Computer Science*. 2005, LNCS (Springer). 3618:224-235
61. Barthélemy JP, Guénoche A: *Trees and proximity relations*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Chichester, 1991
62. Gascuel O: **A note on Sattath and Tversky's, Saitou and Nei's and Studier and Keppler's algorithms for inferring phylogenies from evolutionary distances.** *Mol. Biol. Evol.* 1994, 11(6): 961-963
63. Charleston M, Hendy M, Penny D: **Neighbor-joining uses the optimal weight for net divergence.** *Mol. Phyl. Evol.* 1993, 222:6-12
64. Bryant D: **On the uniqueness of the selection criterion in neighbor-joining.** *J. Classif.* 2005, 22:3-15
65. Sanjuán R, Wróbel B: **Weighted least-squares likelihood ratio test for branch testing in phylogenies reconstructed from distance measures.** *Syst. Biol.* 2005, 54(2): 218-229

Figure legends

Figure 1 - Topological accuracy depending on the rate of missing entries

Horizontal axis: percentage of missing distances (P_{miss}). Vertical axis: topological accuracy measured by the mean (over 500 trials) quartet distance (d_q) between the correct and inferred trees. s : number of taxon pairs that BIONJ* first selects using NJ-like Q_{xy}^* criterion (6), and then analyzes using score-based N_{xy}^* criterion (9) (and criteria (8), (10), (11) in case of ties). The distance matrix is additive, and thus all methods recover the correct tree when $P_{miss} = 0$.

Tables and Captions

Table 1 - Topological accuracy with medium-level distance supermatrices

(a): 25% taxon deletion rate						
<i>k</i> =	FITCH	MW*	NJ*	BIONJ*	MVR*	<i>p</i> -value
2	0.0841	0.0906	0.0926	<u>0.0841</u>	0.0857	0.286
4	0.0504	0.0546	0.0595	<u>0.0494</u>	0.0524	0.466
6	0.0400	0.0445	0.0454	<u>0.0370</u>	0.0410	0.015
8	0.0330	0.0356	0.0386	<u>0.0318</u>	0.0320	0.958
10	0.0271	0.0300	0.0317	<u>0.0265</u>	0.0286	0.364
12	0.0294	0.0317	0.0354	<u>0.0284</u>	0.0314	0.030
14	0.0245	0.0266	0.0286	<u>0.0235</u>	0.0251	0.816
16	0.0290	0.0318	0.0327	<u>0.0282</u>	0.0303	0.028
18	0.0252	0.0278	0.0280	<u>0.0234</u>	0.0265	0.020
20	0.0242	0.0259	0.0281	<u>0.0230</u>	0.0247	0.955

(b): 75% taxon deletion rate						
<i>k</i> =	FITCH	MW*	NJ*	BIONJ*	MVR*	<i>p</i> -value
2	0.2154	0.2174	0.2187	<u>0.2131</u>	0.2163	0.920
4	<u>0.1683</u>	0.1778	0.1818	0.1713	0.1713	0.060
6	<u>0.1347</u>	0.1443	0.1534	0.1418	0.1400	≈ 0.0
8	<u>0.1089</u>	0.1253	0.1302	0.1137	0.1114	0.176
10	<u>0.0878</u>	0.1039	0.1117	0.0959	0.0901	0.033
12	<u>0.0825</u>	0.0968	0.1021	0.0875	0.0842	0.470
14	<u>0.0652</u>	0.0749	0.0850	0.0710	0.0676	0.464
16	<u>0.0583</u>	0.0731	0.0802	0.0658	0.0625	0.335
18	<u>0.0516</u>	0.0617	0.0687	0.0552	0.0555	0.074
20	<u>0.0503</u>	0.0600	0.0682	0.0560	0.0509	0.189

In the medium-level combination, distance matrices are directly estimated from each of the genes and then combined (using SDM) into the distance supermatrix. Topological accuracy is measured by the mean (over 500 trials) quartet distance (d_q) between the correct and inferred trees. *k*: number of genes. *p*-value: sign-test significance when comparing the 500 d_q values of the two best methods that are indicated in bold and underlined (1st method) and bold (2nd one).

Table 2 - Topological accuracy with high-level distance supermatrices

(a): 25% taxon deletion rate						
<i>k</i> =	FITCH	MW*	NJ*	BIONJ*	MVR*	<i>p</i> -value
2	0.0558	0.0561	0.0586	0.0566	<u>0.0522</u>	≈ 0.0
4	0.0337	0.0345	0.0361	0.0351	<u>0.0319</u>	≈ 0.0
6	0.0253	0.0265	0.0272	0.0261	<u>0.0235</u>	≈ 0.0
8	0.0227	0.0228	0.0213	0.0217	<u>0.0212</u>	0.094
10	0.0187	0.0188	0.0194	0.0192	<u>0.0171</u>	0.047
12	0.0197	0.0207	0.0215	0.0199	<u>0.0191</u>	0.949
14	<u>0.0160</u>	0.0164	0.0164	0.0165	0.0162	0.882
16	0.0208	<u>0.0204</u>	0.0210	0.0213	0.0206	≈ 0.0
18	<u>0.0170</u>	0.0177	0.0177	0.0173	0.0174	0.271
20	0.0162	0.0168	0.0171	0.0160	<u>0.0158</u>	0.648

(b): 75% taxon deletion rate						
<i>k</i> =	FITCH	MW*	NJ*	BIONJ*	MVR*	<i>p</i> -value
2	0.1876	0.1877	0.1824	0.1822	<u>0.1817</u>	0.282
4	0.1396	0.1397	0.1390	0.1381	<u>0.1345</u>	0.018
6	0.1095	0.1125	0.1134	0.1119	<u>0.1065</u>	0.166
8	0.0865	0.0892	0.0926	0.0870	<u>0.0823</u>	0.005
10	0.0690	0.0739	0.0766	0.0723	<u>0.0671</u>	0.023
12	0.0641	0.0670	0.0705	0.0677	<u>0.0616</u>	0.015
14	0.0508	0.0538	0.0567	0.0534	<u>0.0493</u>	≈ 0.0
16	0.0504	0.0518	0.0554	0.0512	<u>0.0457</u>	≈ 0.0
18	0.0409	0.0416	0.0485	0.0424	<u>0.0402</u>	0.922
20	0.0403	0.0435	0.0453	0.0431	<u>0.0371</u>	≈ 0.0

In the high-level combination, ML trees are first inferred separately for every genes, and then these trees are turned into path-length distance matrices which are combined (using SDM) into the distance supermatrix. For symbols and notation, see note to Table 1.

Table 3 - Topological accuracy with datasets generated from Driskell et al. [20]

(a): medium level							
FITCH	MW*	NJ*	BIONJ*	MVR*	<i>p</i> -value		
					MVR* - FITCH	MVR* - BIONJ*	
0.0234	0.0268	0.0289	0.0227	<u>0.0171</u>	≈ 0.0	≈ 0.0	

(b): high level							
FITCH	MW*	NJ*	BIONJ*	MVR*	MRP	<i>p</i> -value	
						MVR* - FITCH	MVR* - MRP
0.0161	0.0165	0.0182	0.0172	<u>0.0101</u>	0.0119	0.001	0.193

(a): Medium-level combination of the distance matrices being directly estimated from the gene sequences. (b): High-level combination; ML trees are first inferred separately for every genes; MRP turns the gene trees into a matrix of partial binary characters, which is analyzed with parsimony; with the other (distance) methods, the gene trees are turned into path-length distance matrices which are combined into the distance supermatrix. All combinations of source distance matrices are achieved using SDM. *p*-value: sign-test significance when comparing the 100 d_q values of MVR* (our best algorithm) to those of FITCH and MRP. For other symbols and notation, see note to Table 1.

Table 4 - Run times

(a): 25% taxon deletion rate										
	SDM		FITCH			MW*		MVR*		
	<i>k</i> =		<i>k</i> =			<i>k</i> =		<i>k</i> =		
	10	2	10	20	2	10	20	2	10	20
<i>n</i> = 48	< 1	11	23	23	21	39	41	< 1	< 1	< 1
<i>n</i> = 96	5	437	482	479	623	932	926	7	6	5
<i>n</i> = 192	32	11,065	13,864	13,945	23,541	34,368	35,017	57	60	42

(b): 75% taxon deletion rate										
	SDM		FITCH			MW*		MVR*		
	<i>k</i> =		<i>k</i> =			<i>k</i> =		<i>k</i> =		
	10	2	10	20	2	10	20	2	10	20
<i>n</i> = 48	< 1	6	17	23	10	28	36	< 1	< 1	< 1
<i>n</i> = 96	< 1	22	455	492	29	656	667	< 1	4	7
<i>n</i> = 192	2	448	11,532	14,025	916	32,371	34,152	3	31	52

(c): 1.2% missing distances (Driskell et al.)										
	SDM		FITCH			MW*		MVR*		
	<i>k</i> = 254		<i>k</i> = 254			<i>k</i> = 254		<i>k</i> = 254		
<i>n</i> = 48	334		132			268			< 1	

Mean run times are provided for various taxon numbers (*n*), gene numbers (*k*) and proportions of missing entries: (a) 25%, (b) 75%, and (c) 1.2% using Driskell et al. [20]-like datasets. Run times are measured in seconds using a standard PC (Pentium IV 1.8GHz, 1Gb RAM). The low run times with *k* = 2 and 75% deletion rate are explained by the low size of the distance super-matrices (see text for explanations).