國 立 政 治 大 學 資 訊 科 學 系
Department of Computer Science
National Chengchi University

碩 士 學 位 論 文
Master's Thesis

BigBigTree: 基於 Nextflow 框架利用分群串接法建立巨量同源基因演化樹

BigBigTree: a divide and concatenate strategy for the phylogenetic reconstruction of large orthologous datasets using Nextflow framework

指導教授：張家銘 博士

研究生：蔡漢龍 撰

中 華 民 國 109 年 7 月

DOI:10.6814/NCCU202001688

謝辭

能完成這本論文要感謝我的指導教授張家銘老師，謝謝您對於論文逐字逐句地檢閱，以及在研究碰到難題時給我提供許多靈感。謝謝實驗室的各位同學平時給予我的鼓勵與幫助，特別感謝我的大學同學張芷銓，協助我完成 BigBigTree 的開發及設計。

感謝論文口試委員吳育瑋老師、蔡怡陞老師，謝謝兩位老師花時間閱讀論文初稿，不僅指出錯誤，還給予許多建設性的建議，讓我能把論文做的更完善。還要感謝《財團法人國家實驗研究院‧國家高速網路與計算中心》提供軟硬體資源，使本研究得以順利進行。

最後要感謝我的家人，謝謝您們從小到大對我的支持，讓我不用擔心太多事情，順利地完成學業。

DOI:10.6814/NCCU202001688

摘要

演化樹（phylogenetic tree）是根據不同生物間的型態、構造、生理、生態、遺傳和基因序列等特徵，將生物做系統化的分類，做成各物種間演化、親緣關係的樹狀圖，從中我們可以了解到序列間推斷的演化歷史。由於次世代定序技術及第三代定序技術的發展，越來越多的基因資料可以取得，面對龐大的資料量，甚至是最快的方法都具有挑戰性。一些重要的多基因家族(如嗅覺受體)已無法通過最準確的方法—最大似然(Maximum likelihood）來構建系統發育樹。

在本研究中，我們提出了 BigBigTree，透過分群串接法將問題分解為較小的問題並獨立解決。這個方法依賴於在直系同源基因的大型數據集中，進行分群的能力，每群直系同源基因都使用一種典型方法來構建演化樹，並在第二階段處理樹的上層(超級樹)，從每棵子樹中選擇每種物種的一種蛋白質序列，對來自同一物種的所有蛋白質序列進行多重序列比對，最後依其直系同源關係將序列串接起來，用於建構超級樹。這個方法的優點是我們減少了要分類的序列數量，且不會丟失資訊，因為最後的串接序列代表所有的序列。BigBigTree 可以有效地處理特定於譜系的重複，但不能處理基因水平轉移，它更適合分析大的真核生物家族，如激酶或嗅覺受體。

我們利用真實數據及模擬數據對 BigBigTree 進行評估，並與 RAxML v8.2.12、RAxML-ng 及 IQ-TREE2 比較結果。在大多數情況下，BigBigTree 的執行時間比 RAxML 和 RAxML-ng 快。在拓樸精度方面，BigBigTree 在模擬數據上展現比其他方法更好的性能，並在實際數據中獲得與其他方法接近的精度。BigBigTree 的原始碼及 docker 容器可在 https://github.com/jmchanglabtw/bigbigtree 和 https://hub.docker.com/r/changlabtw/bigbigtree 中取得。

關鍵字：基因樹、演化樹、Nextflow、分群串接

DOI:10.6814/NCCU202001688

Abstract

A phylogenetic tree is a branching diagram based on the similarities of creatures in morphology, structure, physiology, genetics, ecology, and genetic sequence. It shows an inferred evolutionary history among sequences. Thanks to the next-generation sequencing technique and the third-generation sequencing technique, more and more sequences have become available. This overwhelming amount of data is challenging, even the fastest methods. Some important multi-genetic families like olfactory receptors have become impossible to build a phylogenetic tree with the most accurate methods like Maximum Likelihood (ML).

Here we show how a simple Divide and Concatenate strategy, BigBigTree, can be applied to this problem by breaking it down into smaller problems that are solved independently. Our approach relies on the ability to identify within large dataset clusters of orthologous genes. Each group of orthologous genes is used to build a phylogenetic tree using a typical approach. The upper level of the tree (super-tree) is resolved in a second stage. One protein per species is chosen from each subtree. All proteins from the same species are aligned together. The alignment used for building the super-tree results from concatenating all these alignments, where within-species paralogues appear in the same columns, and orthologues appear in the same row. The advantage is that we reduce the number of sequences to classify without losing information as all sequences are represented in the final alignment. This approach can efficiently deal with lineage-specific duplications, but not with lateral transfers. It is better suited for the analysis of large eukaryotic families like the kinases or the olfactory receptors.

We evaluated BigBigTree in simulation and real data sets against RAxML v8.2.12, RAxML-ng, and IQ-TREE2. BigBigTree is faster than RAxML and RAxML-ng in most cases. Regarding topology accuracy, BigBigTree shows better performance than others in simulation data and gets compatible accuracy with others in real data. The source code and docker of the method are available at https://github.com/jmchanglabtw/bigbigtree and https://hub.docker.com/r/changlabtw/bigbigtree, where the latter allows users one-click installation.

Keywords: gene tree, phylogenetic tree, Nextflow, divide and concatenate

DOI:10.6814/NCCU202001688

Contents

DOI:10.6814/NCCU202001688

List of Tables

DOI:10.6814/NCCU202001688

List of Figures

DOI:10.6814/NCCU202001688

1 Introduction

1.1 Research motivation

Phylogenetic tree classifies creatures by their genetic similarities. It can facilely find the common ancestors of every species. Take Figure 1 as an example, each node represents the nearest common ancestor of each branch, and the edge lengths in phylogenetic trees may be interpreted as time estimates.

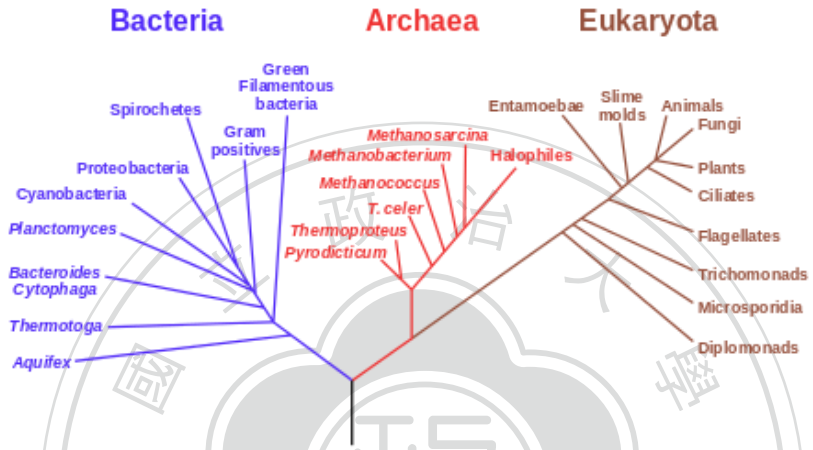


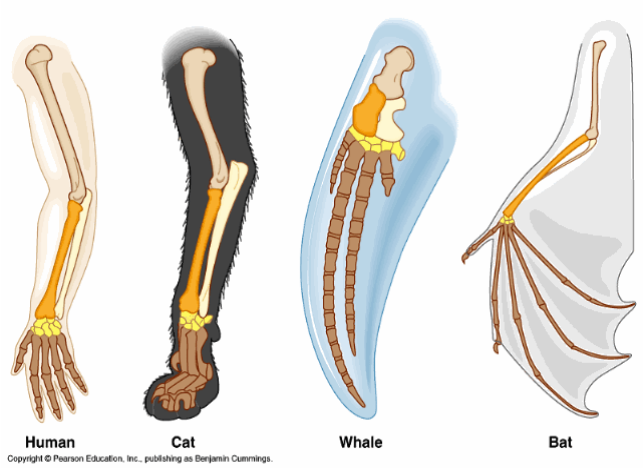
Figure 1：Phylogenetic Tree of Life [1]

Building a phylogenetic tree needs to confirm the relevance between species since evolution is an extremely time-consuming process. We can not prove it through observation or experiment directly. Instead, we only show it by collateral evidence, that means all of the phylogenetic trees are hypotheses, building phylogenetic trees by different models and methods may produce different results. The major of biologists use two types of ways as the basis to confirm the similarities of species.

First is morphology, the characterizations of species fall into homology and analogy. Homology means creatures have resembled body structures, but evolve into different appearances and abilities that depend on their living environment (Figure 2.a). Analogy, precisely the opposite, having similar presentations or skills but grow from different body structures. The analogy may not have a genetic relationship, only the result of convergent evolution (Figure 2.b). Therefore, if we want to use the similar characterizations between fossils and living creatures to confirm their relevance, we must base it on the homology to ensure the close and distant relationships of species.

DOI:10.6814/NCCU202001688

(a)



(b)

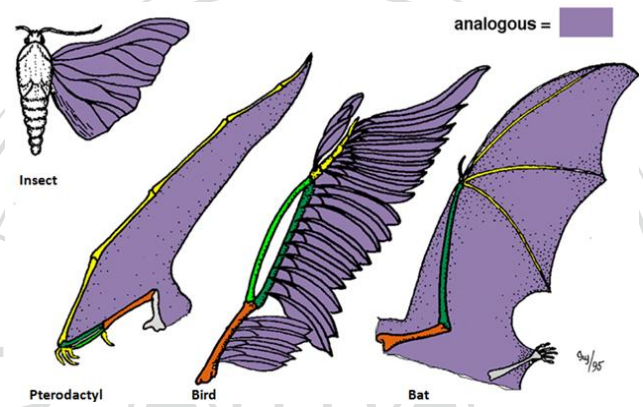


Figure 2：(a) Homology and (b) Analogy [2].

The other one is molecular biology, using DNA sequencing, which confirms relevance between creatures by order of four bases—adenine (A), guanine (G), cytosine (C), and thymine (T) —in a strand of DNA.

A phylogenetic tree shows the close and distant relationships of species. It can handily classify each genotype and find the needed parts that can be used in genetic modification and identification. There are many ways to build a phylogenetic tree. Still, most of them could make trees quickly only when the input genetic data is small. When face to vast amounts of data, how to optimize the process of building a phylogenetic tree became an issue. To construct a phylogenetic tree accurately and rapidly, we have two popular methods: Maximum Parsimony (MP) and Maximum Likelihood (ML).

MP is using the degree of variation between genetic sequences to confirm the close and distant relationships of species, that is, building a phylogenetic tree by changing the least in the evolution, this way may misclassify different creatures into similar species caused by the

DOI:10.6814/NCCU202001688

convergent evolution. Therefore, this way usually uses in the situation that the relationships of species are close; MP is the way that is using statistical data to estimate the stochastic model.

Thanks to the development of Next-Generation Sequencing and Third-Generation Sequencing, we can get lots of genetic sequences quickly. It is capable of building a phylogenetic tree by genetic sequence. Compared to MP, which is easy to understand but has more deviation, ML is more accurate but needs enormous calculations. Our research hopes to combine the advantages of MP and ML, using the Divide and Concatenate algorithm. It clusters genetic sequences by their homology, disposing of them separately, and then merge. The algorithm not only keeps the complicated genetic information ultimately but also can build a phylogenetic tree accurately and quickly.

1.2 Related works

S. Mirarab et al. presented that although there are several multispecies coalescent models, they all have disadvantages. BUCKy-pop has high time complexity, even if it can use an unrooted tree [3]. BEST and *BEAST can build gene trees and species trees by sequence alignment simultaneously, but when data is big enough will lead those methods limitless [4]. Another thesis also presented conflicts that may occur if it uses different alleles to build different phylogenetic trees. To correct these conflicts, it needs to align several alleles to increase the accuracy of phylogenetic trees, making the data that it needs enormous [5]. Thus, the problem in front of the development of genetic technology is to reduce time complexity. Accurate Species TRee ALgorithm (ASTRAL) in the thesis is a way limit searching space by abandoning the less grande side to make time complexity in polynomial time; in the other way, we reduce time complexity by dividing and merging data [4,5].

P. Vachaspati *et al.* presented that many methods use ILS (Incomplete Lineage Sorting) in building species trees. Still, only ASTRAL-2 and NJst can remain accurate in a massive data level, so he redesigns NJst to improve the compatibility of data. And it combines different distance-based tree estimative methods, called ASTRID, with similar accuracy and even better efficiency than ASTRAL-2 [6]. The thesis also mentioned that INternode Distances is essential in building phylogenetic trees, calculate the period in the evolution process to analyze relationships, and make the correspondent time be the proportion of the distance between nodes in phylogenetic trees, which can make the results more accurate.

Although there are many large-scale tree reconstruction methods, deep branches of the tree tend to have very low supports and weak evolutionary signals [7]. One popular approach to increase the evolutionary signal is super-matrix, which concatenates gene families [8]. It has

DOI:10.6814/NCCU202001688

been shown to resolve the Yeast Tree of Life based on the concatenation of 20 genes [9]. Like super-matrix, Ashkenazy, H. *et al.* [10] and Chang *et al.* [11] independently propose super-MSA, a concatenation of different MSAs of the same input. Ashkenazy, H. *et al*. found the tree reconstructed from the super-MSA is more accurate than one by an individual MSA. However, Chang *et al.* found that of the super-MSA tree is as good as one of the MSA. Interestingly, they showed the bootstrap of the super-MSA is more informative than one based on an MSA. Taking concatenation a step further, we propose BigBigTree, a strategy combining computational Divide and Conquer algorithm and concatenation to resolve the low confidence in the deep node of the large orthologous gene family.

DOI:10.6814/NCCU202001688

2 Methods

2.1 Our Idea

In the course of meiosis or RNA replication, it will occur a phenomenon called gene duplication, a duplication of the gene region. Gene duplications are an essential source of genetic novelty that can lead to evolutionary innovation. Duplication creates genetic redundancy, where the extra copy of the gene is often free from selective pressure. If one copy of the gene experiences a mutation affecting its original function, other copy can serve as a 'spare part' and continue to function correctly. A variation will have no harmful effects on its host organism. Thus, duplicate genes accumulate mutations faster than a functional single-copy gene, which is recognized as an evolutionary manifestation.

According to Figure 3, when the genome region of an ancient species occur gene duplication, it produces α-gene and β-gene in descendant species. Afterward, with a speciation event, the species gradually evolve into three species frag, chick, and mouse. The α-genes of three species are ‘orthologous genes.’ The α-gene and β-gene of the same species are ‘paralogous genes.’

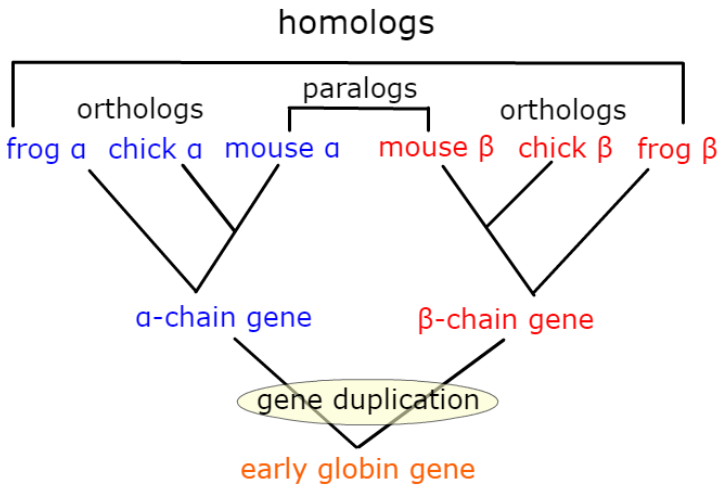


Figure 3：Difference between orthology and paralogy.

Orthologous genes are more similar than paralogous genes because the later had mutated before speciation compared to the former had mutated after speciation. So, the orthologous genes between different species are more similar than the paralogous genes of the same species. Our approach mainly takes advantage of this feature. First, sequences are clustered into groups (i.e., orthologous genes) used to build individual trees. Then, the hierarchical clustering of those groups is determined by the concatenated orthologous alignment. The final phylogenetic tree is constructed by merging single orthologous trees within the hierarchical clustering.

DOI:10.6814/NCCU202001688

2.2 Algorithm Design

The algorithm consists of six steps: 1) input data, 2) use BLAST to compare sequences; 3) cluster sequences based on similarity in the previous step, 4) align each group, 5) and then concatenate each ortholog among species alignment into on single string, 6) finally build a mother tree base on the result of the previous step and merge subtrees of each cluster and the mother tree. The overall flowchart is shown in Figure 7. The detail of each level is explained as the following:

Step1：Input

The input of BigBigTree is the gene sequences with species annotation in FASTA format. For example, there are *m* species and *n* orthology gene families, in total, *m***n* sequences.

Step2：Sequence comparison

We compare *m***n* sequences by the Basic Local Alignment Search Tool (BLAST). BLAST is a program that has been used widely to align the primary structure of biological sequences in analyzing bioinformatics, which can let researchers find target sequences or similar ones, using a heuristic algorithm to search and have quite a speed and accuracy [12]. The time complexity is $O(m^2n^2)$ for all-versus-all BLAST.

Step3：DIVIDE－Cluster ortholog

According to the result of BLAST, we get similarity between sequences. Then, we apply a tool, *hcluster*, cluster sequences into cross-species orthologues, and transfer them into FASTA format [13]. FASTA is a text format used in recording nucleic acid or peptide sequences. Any nucleic acid and amino acid present as a single alphabet code so that we can quickly analyze sequences with a scripting language such as Python, Ruby, and Perl. For example, there are 12 sequences in total cross three *Drosophila* species. We get four ortholog clusters: $cluster_1$, $cluster_2$, $cluster_3$, and $cluster_4$ after the Divide step (Figure 4).

DOI:10.6814/NCCU202001688

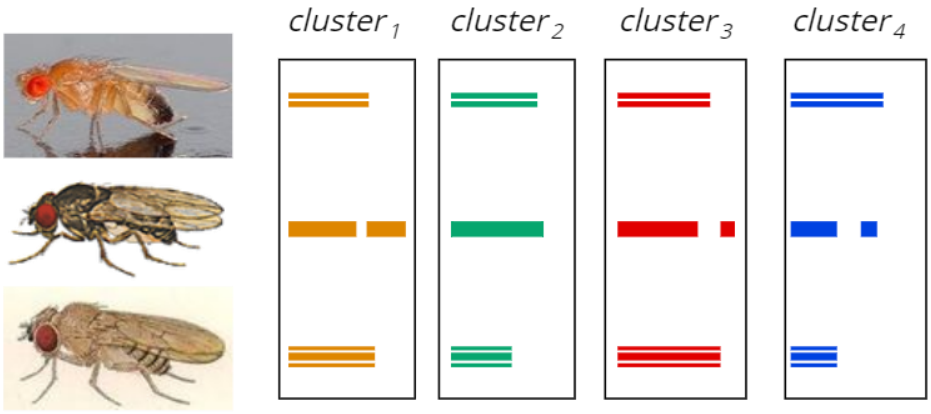


Figure 4：Four ortholog clusters and their corresponding alignments.

Besides, we cluster sequences according to species. For the above example, it results in three clusters: spe_1, spe_2, and spe_3, where each contains paralog sequences (Figure 5).

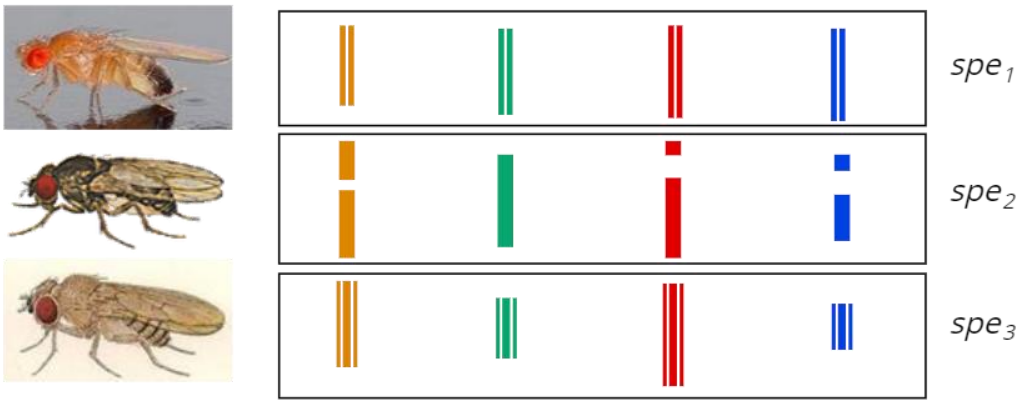


Figure 5：Three species cluster and their corresponding alignments.

Step4：Alignment

Instead of generating a big alignment of all sequences, we use *T-Coffee* [14] or MAFFT [15] to generate alignment for each cluster - ortholog cluster $\{cluster_1, cluster_2, \ldots, cluster_n\}$ (an example in Figure 4) and species cluster $\{spe_1, spe_2, \ldots, spe_m\}$ (an example in Figure 5). It reduces the time complexity to n*ALIGN(m) + m*ALIGH(n) instead of the original ALIGN(m*n) without divide, where ALIGN(x) represents the running time to perform multiple sequence alignment for x sequences. If there is more than one sequence from the same ortholog cluster in a species cluster, BigBigtree will calculate pairwise sequence similarity and then extract the representative sequence by similarity. For example, there are two sequences c_2 and c_3 for species C and c_2 is picked as a representative sequence (Figure 13.c). In contrast, if there is no sequence from one of the ortholog clusters in the species cluster, it will append a gap

DOI:10.6814/NCCU202001688

sequence. For example, gap sequences are filled for species *a* in cluster$_3$ and species *b* and *c* in *cluster*$_2$ (Figure 13.c).

Step5：CONCATENATE－Orthology concatenation

We concatenate each ortholog among species alignment (*spe*$_1$, *spe*$_2$, …, *spe*$_m$) into one string, such as Figure 6. Therefore, we will have the sequence alignment of paralogous strings, which provides more information for phylogenetic reconstruction and reduces time complexity in building an evolutionary tree.

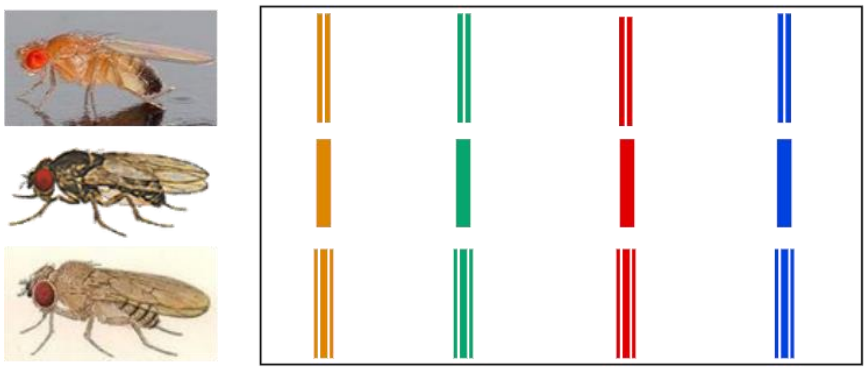


Figure 6：Concatenate long-string alignment from Figure 5.

Step6：Tree reconstruction

Users can choose TreeBeST [16], IQ-TREE2 [17,18], or PhyML [19] to build phylogenetic trees of orthologous sequences alignment - *shallow tree* (i.e., *cluster*$_n$) and the concatenate long-string alignment - *deep tree*. TreeBeST is mainly designed for building gene trees with a known species tree and is highly efficient and accurate. If there is no species tree for input data, we recommend users choose IQ-TREE2 and PhyML instead. Without a species tree, TreeBeST is nothing but a common PhyML or even worse.

Step7：Merge

We construct a final phylogenetic tree by replacing the leaf node of the concatenate tree (deep tree) with corresponding orthologous trees (shallow trees).

Compared with a traditional way, building a tree based on all sequences, *Divide and Concatenate* approach utilizes more information of the concatenated strings to complete a phylogenetic tree accurately. It also reduced time complexity to n*TREE(m) + m*TREE(n), instead of the original TREE(m*n). TREE(x) represents the running time to build a phylogenetic tree for x sequences.

DOI:10.6814/NCCU202001688

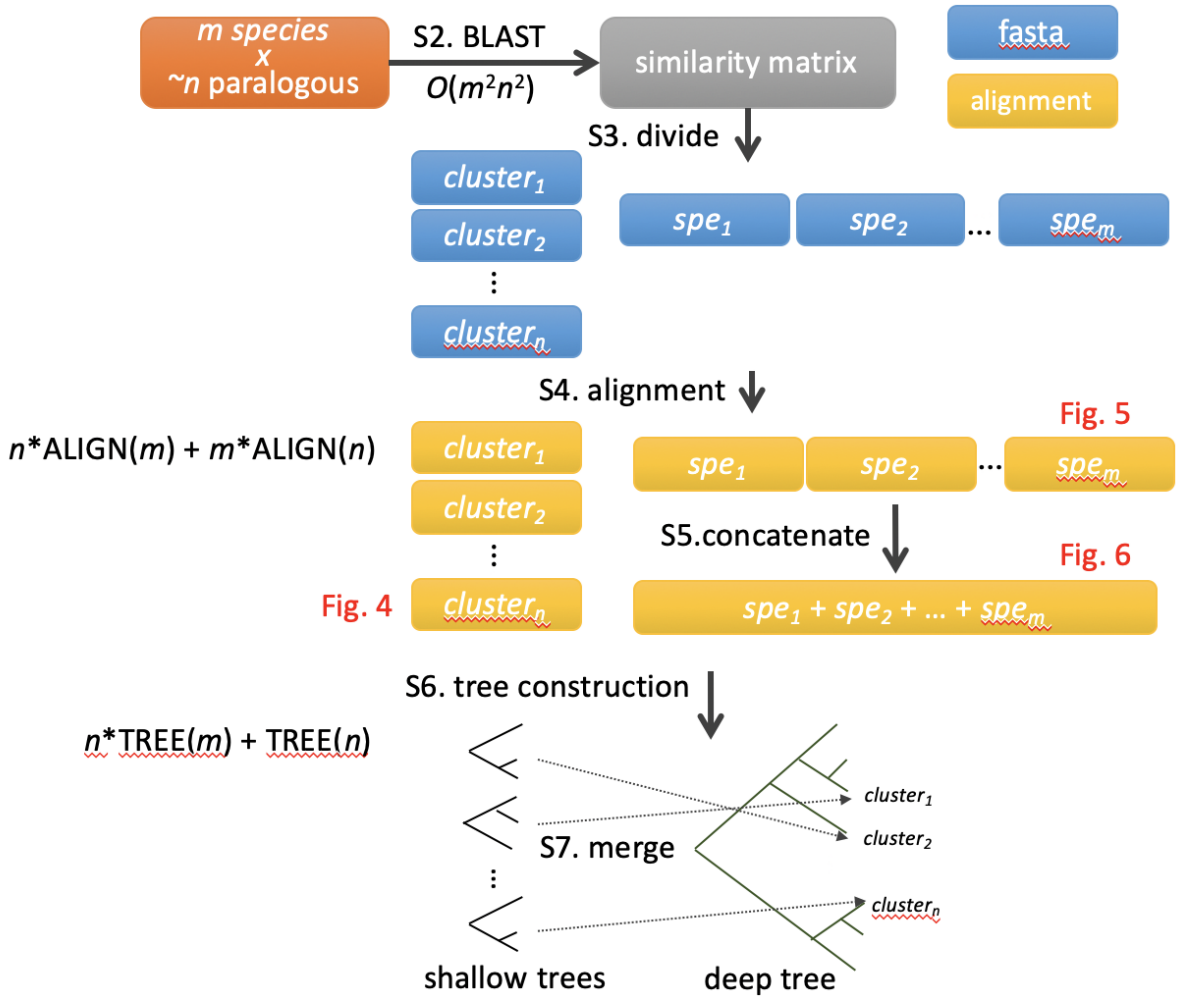


Figure 7：The flow chart of our algorithm in which ALIGN(x) and TREE(x) represent the running time to perform multiple sequence alignment and build a phylogenetic tree for x sequences. The exact running time of those two functions depends on which package is used. The example of each step is marked as red.

2.3 Implementation

We reimplement the whole pipeline by *Nextflow* [20] instead of Python [21]. Nextflow simplifies the implementation and the deployment of complex parallel and reactive workflows used in our project to make the pipeline more quickly and more efficiently (Figure 8). Besides, it can easily allow users to select different methods for multiple sequencing alignment and tree construction (Figure 9). It is possible to execute locally by cloning the repository from GitHub or downloading a docker container where both are available at https://github.com/jmchanglab/bigbigtree and https://hub.docker.com/r/changlabtw/bigbigtree, respectively. We will reconstruct a new web service for biologists without installation.

DOI:10.6814/NCCU202001688

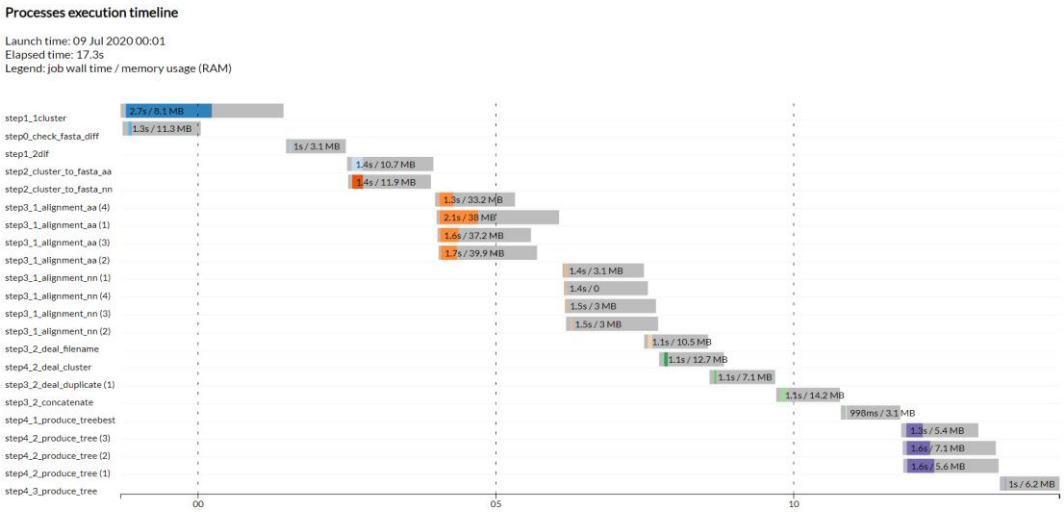


Figure 8：The snapshot of the execution timeline of BigBigTree by Nextflow.

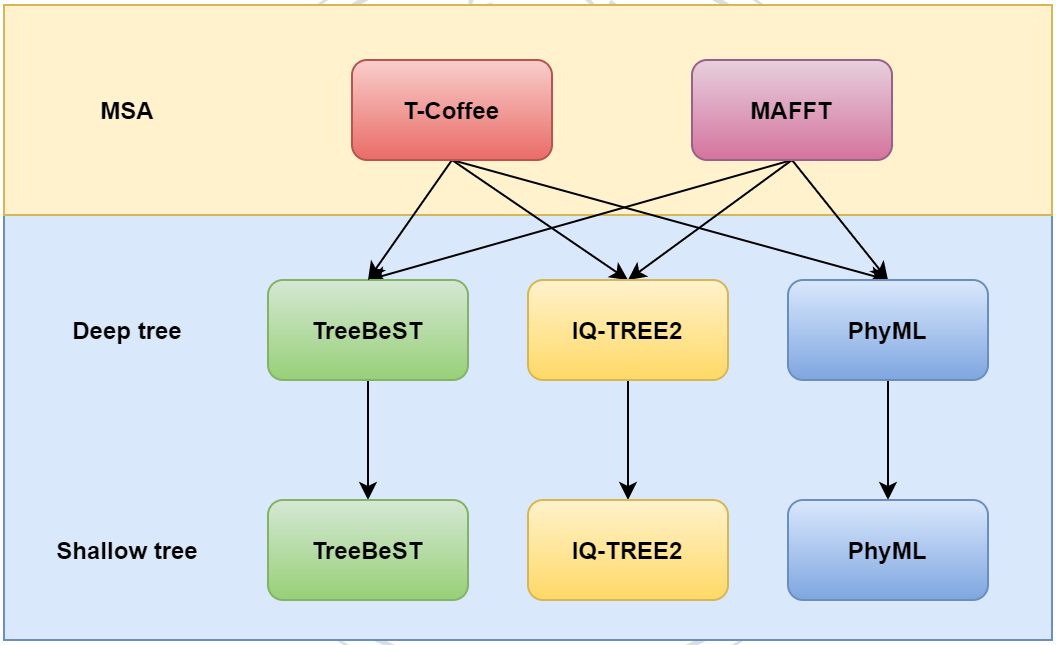


Figure 9：The snapshot of BigBigtree running options for MSA and tree reconstruction.

2.4 Evaluation

2.4.1 Real biology data

For the testing ability of BigBigTree, we collected data sets with two evolutionary scenarios, which are summarized in Table 1.

- Same family size among all species: The data is taken from the JGI web server [22].
- Different family size among all species: Olfactory receptors come from 12 *Drosophila* species.

DOI:10.6814/NCCU202001688

After that, besides running time, we evaluated the quality of the tree by calculating its log-likelihood score by IQ-TREE2 with topology constraint. Taking the alignment sequences and the user tree as input, IQ-TREE2 will perform several tree topology tests and the expected likelihood weight. Thus, we can get the log-likelihood score of the user tree. IQ-TREE2 command is the following where *result.aln* is MSA of the input sequences, and *result.tree* is a tree for evaluation.

```
./iqtree2 -s <result.aln> -m TESTONLY -te <result.tree>
```

Table 1. The summary of real data.

Data Set type	# of gene families	# of species	# of sequences
Same family size			
Cladiomy	5	3	15
microspor	29	8	232
Different family size			
Olfactory receptor	~60	12	858

2.4.2 Simulation biology data

Because real phylogenetic trees are hypotheses, there is no correct answer to evaluate the accuracy of the result. To overcome the difficulty, we generate simulation data sets using *Simphy* [23] and *INDELible* [24]. The simulation flowchart is shown in Figure 10.

DOI:10.6814/NCCU202001688

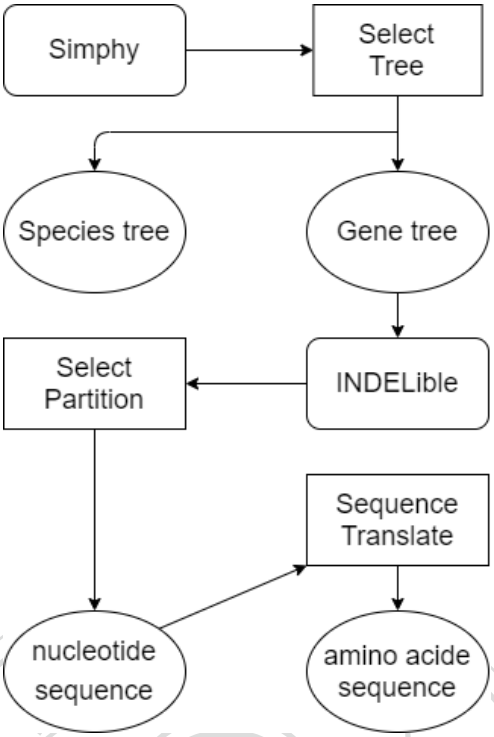


Figure 10：The diagram of the simulation data.

Simphy is used for generating a species tree with a set of gene trees evolving within the species trees. Because Simphy incorporates speciation and duplication events in its simulations, the true relation of orthologs and paralogs are known. It helped us to analyze the accuracy of the orthologs cluster in BigBigTree. We generated five species trees and 50 gene trees, ~ ten gene families, for each species tree, then we selected a gene tree for each species tree as the simulation data. The Simphy parameters referred to the settings in the HyPPO simulation experiment [25]. The species tree height (time units) was set to 50000000 to ensure that it has enough time to evolve more genes. The duplication and loss rates were set to 0.0000005. The transfer rate was set to 0.

We then used INDELible to simulate the evolution of gene sequences on each gene tree. The evolve number was set to 10 for generating ten replicate datasets. After that, we selected a dataset that it didn't contain null sequences as final simulation data. To evaluate the impact of evolutionary time, we used the *treedepth* parameter to rescale the tree to have a maximum root-to-tip distance of 0.5, 1, and 1.5. Otherwise, The codon type evolved under the M0 model. The insertion and deletion rate was set to 0.1. Output files were nucleotide coding gene sequences, and we translated them into protein-coding gene sequences afterward. Table 2 summary of the simulation data.

DOI:10.6814/NCCU202001688

Table 2. The summary of simulation data for treedepth 0.5, 1 and 1.5

data set	# of species	# of gene families	# of sequences
5x8	5	~8	36
5x25	5	~25	118
9x9	9	~9	75
9x20	9	~20	176
9x33	9	~33	302

Thus, we can compare outputs against ground truths based on *Robinson Foulds* (RF) distance metric [26]. RF distance is a generalization of the same metric for phylogenetic trees. Briefly, it is half of the cardinality of the symmetric difference of the two sets of clusters respectively contained in the two networks [27]. It sums up the number of partitions of data implied by the first tree but not the second tree and the number of partitions of data involved by the second tree but not the first tree. The higher the distance between the output tree and the right tree, the higher the difference between the two trees.

2.5 Comparison

We compared BigBigTree against RAxML v8.2.12 [28], RAxML-ng [29], and IQ-TREE2, which are programs for sequential and parallel Maximum-Likelihood based inference of large phylogenetic trees. All evaluations are conducted in TWCC (Taiwan Computing Cloud) virtual compute service with 2.4 GHz, four-core CPU, and 32 GB memory under Ubuntu Linux. Since there are 12 versions of RAxML, we choose raxmlHPC-PTHREADS-AVX as the opponent. It exploits the capabilities of the CPU(s) well (Figure 11).

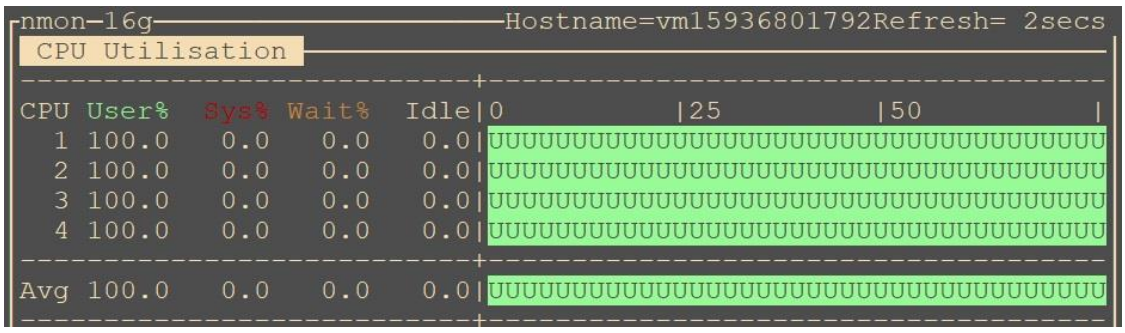


Figure 11：The snapshot of CPUs running with raxmlHPC-PTHREADS-AVX

DOI:10.6814/NCCU202001688

RAxML conducts a rapid bootstrap analysis and searches for the best-scoring ML tree. Because the number of bootstraps strongly influences the execution time, we execute RAxML with one bootstrap and 100 bootstraps. Similarly, RAxML-ng and IQ-TREE2 also have rapid options. Then we execute rapid mode and standard mode, respectively. After that, phylogenetic trees were reconstructed using RAxML with the PROTCATDAYHOFF substitution model, RAxML-ng with the Jones-Taylor-Thornton (JTT) + G model [30], and IQ-TREE2 with the JTT + F +G model. The model selection of RAxML-ng and IQ-TREE2 referred to the TCS phylogenetic benchmark [31]. These execution commands are shown below:

RAxML v8.2.12

raxmlHPC-PTHREADS-AVX -T 4 -f a -x 12345 -N 1 -m PROTCATDAYHOFF -s input.fasta -p 12345

raxmlHPC-PTHREADS-AVX -T 4 -f a -x 12345 -N 100 -m PROTCATDAYHOFF -s input.fasta -p 12345

RAxML-ng

raxml-ng --search1 --msa input.fasta --model JTT+G --thread 4 --seed 2

raxml-ng --msa input.fasta --model JTT+G --thread 4 --seed 2

IQ-TREE2

iqtree2 -s input.fasta -m JTT+F+G -fast -T 4

iqtree2 -s input.fasta -m JTT+F+G -T 4

Thus, we compared BigBigTree to RAxML-1-bootstrap, RAxML-ng-search1, and IQ-TREE2-fast for execution time; RAxML-100-bootstrap, RAxML-ng-standard, and IQ-TREE2-standard for RF distance. Additionally, since TreeBeST, which is used to build phylogenetic trees in BigBigTree, invokes `best' command, the resultant tree is bootstrapped 100 times. The input data RAxML, RAxML-ng, and IQ-TREE2 needed were alignment data. We used MAFFT to align raw data into alignment data to build phylogenetic trees with them and then compared the results to BigBigTree.

DOI:10.6814/NCCU202001688

3 Result

3.1 Simulation biology data

We first use the simulation data to evaluate BigBigTree with multiple sequence alignment methods and tree construction methods. The running time comparison is summarized in Table 3 and Table 5. We divide the running time into three parts and analyze it, respectively. The RF-distance comparison is summarized in Table 4 and Table 6. In the aspect of a multiple sequence alignment method, we can observe that the results of T-Coffee and MAFFT are very close. According to the T-Coffee manual, it recommends using T-Coffee when the number of sequences is less than 100. If the number of sequences is more than 100, we use MAFFT. In the aspect of the tree construction method, We can observe that the execution time of TreeBeST and IQ-TREE2 are similar, but TreeBeST is much better than IQ-TREE2 on RF-distance with a known species tree. Then we use TreeBeST as our primary method.

DOI:10.6814/NCCU202001688

Table 3. Average running time (in secs) of the simulated data set with multiple sequence alignment methods in BigBigTree (fasted marked in Bold).

	T-Coffee + TreeBeST			MAFFT + TreeBeST		
set (*spe* x *gene*)	aln	tree	aln+tree	aln	tree	aln+tree
treedepth = 0.5						
5x8	8	15	23	7	12	**19**
5x25	17	37	54	14	38	**52**
9x9	11	19	30	9	18	**27**
9x20	21	63	84	16	59	**75**
9x33	32	122	154	24	120	**144**
treedepth = 1						
5x8	8	15	23	6	14	**20**
5x25	16	44	60	14	39	**53**
9x9	11	20	31	9	19	**28**
9x20	21	65	86	18	62	**80**
9x33	26	98	124	24	99	**123**
treedepth = 1.5						
5x8	8	13	**21**	7	16	23
5x25	20	33	53	16	30	**46**
9x9	13	20	34	10	18	**28**
9x20	22	49	71	19	46	**65**
9x33	35	138	173	26	137	**163**

DOI:10.6814/NCCU202001688

Table 4. RF-distance analysis of the simulated data set with multiple sequence alignment methods in BigBigTree (Best marked in Bold).

	T-Coffee + TreeBeST				MAFFT +TreeBeST			
set (*spe* x *gene*)	test1	test2	test3	ave	test1	test2	test3	ave
treedepth = 0.5								
5x8	18	30	32	**26.6**	18	30	32	**26.6**
5x25	134	124	128	128.6	134	124	126	**128.0**
9x9	74	60	34	**56.0**	74	60	34	**56.0**
9x20	226	214	242	227.3	212	214	246	**224.0**
9x33	524	466	464	484.6	522	462	466	**483.3**
treedepth = 1								
5x8	22	42	24	**29.3**	22	42	24	**29.3**
5x25	118	112	74	**101.3**	126	112	70	102.6
9x9	58	48	102	**69.3**	60	48	100	**69.3**
9x20	196	204	126	175.3	198	200	124	**174.0**
9x33	480	468	284	**410.6**	486	468	284	412.6
treedepth = 1.5								
5x8	46	20	10	**25.3**	46	20	10	**25.3**
5x25	96	72	104	**90.6**	96	72	104	**90.6**
9x9	40	44	44	42.6	40	44	36	**40.0**
9x20	152	100	158	**136.6**	152	100	158	**136.6**
9x33	492	400	372	**421.3**	490	402	372	**421.3**

DOI:10.6814/NCCU202001688

Table 5. Average running time (in secs) of the simulated data set with different tree construction methods in BigBigTree (fasted marked in Bold).

	T-Coffee + TreeBeST			T-Coffee + IQ-TREE2		
set (spe x gene)	aln	tree	aln+tree	aln	tree	aln+tree
treedepth = 0.5						
5x8	8	15	23	8	14	**22**
5x25	17	37	54	17	35	**52**
9x9	11	19	**30**	11	26	37
9x20	21	63	84	21	58	**79**
9x33	32	122	154	32	120	**152**
treedepth = 1						
5x8	8	15	23	8	14	**22**
5x25	16	44	**60**	16	44	**60**
9x9	11	20	**31**	11	21	32
9x20	21	65	**86**	21	65	**86**
9x33	26	98	124	26	93	**119**
treedepth = 1.5						
5x8	8	13	**21**	8	22	30
5x25	20	33	**53**	20	35	55
9x9	13	20	**34**	13	31	44
9x20	22	49	**71**	22	70	92
9x33	35	138	173	35	130	**165**

DOI:10.6814/NCCU202001688

Table 6. RF-distance analysis of the simulated data set with different tree construction methods in BigBigTree (Best marked in Bold).

	T-Coffee + TreeBeST				T-Coffee + IQ-TREE2			
set (spe x gene)	test1	test2	test3	ave	test1	test2	test3	ave
treedepth = 0.5								
5x8	18	30	32	**26.6**	32	47	36	38.3
5x25	134	124	128	**128.6**	170	161	158	163.0
9x9	74	60	34	**56.0**	82	94	82	86.0
9x20	226	214	242	**227.3**	246	250	256	250.6
9x33	524	466	464	**484.6**	524	501	501	508.6
treedepth = 1								
5x8	22	42	24	**29.3**	28	48	40	38.6
5x25	118	112	74	**101.3**	148	146	121	138.3
9x9	58	48	102	**69.3**	81	85	102	89.3
9x20	196	204	126	**175.3**	227	219	150	198.6
9x33	480	468	284	**410.6**	501	489	341	443.6
treedepth = 1.5								
5x8	46	20	10	**25.3**	44	23	26	31.0
5x25	96	72	104	**90.6**	136	108	144	129.3
9x9	40	44	44	**42.6**	69	71	69	69.6
9x20	152	100	158	**136.6**	172	136	186	164.6
9x33	492	400	372	**421.3**	486	458	412	452.0

DOI:10.6814/NCCU202001688

After that, the running time comparison between BigBigTree, RAxML, RAxML-ng, and IQ-TREE2 is summarized in Table 7. We also divide the running time into three parts and analyze it, respectively. In the alignment part, it can be observed that MAFFT is much faster than BigBigTree in the whole process, while BigBigTree is faster than MAFFT in each cluster. In the tree construction part, IQ-TREE2 performs fastest with a quick option. Besides, BigBigTree is faster than RAxML and RAxML-ng at a larger dataset. The execution time of three is similar to a smaller dataset. We owe it all to Divide and Concatenate algorithm, which makes time complexity reduce to n*TREE(m) + m*TREE(n), down from TREE(m*n). It can be found that the larger the data, the faster BigBigTree. Also, when the difference between gene sequences increases, BigBigTree doesn't change a lot on running time compared to RAxML and RAxML-ng.

The RF-distance comparison is summarized in Table 8. The results show that our algorithm has great potential to improve accuracy. BigBigTree is more accurate than others on 5x25, 9x9, and 9x20 in most cases. Moreover, similar results can be achieved on 5x8. Nevertheless, the performance is significantly degraded on 9x33, and we will discuss this further in the discussion section.

DOI:10.6814/NCCU202001688

DOI:10.6814/NCCU202001688

Table 7. Average running time (in secs) of the simulated data set (fasted marked in Bold) where RAxMLv8.2.12 using one bootstrap, RAxML-ng using --search1 option, and IQ-TREE2 using -fast option.

	BigBigTree			RAxMLv8.2.12			RAxML-ng			IQ-TREE2		
set (spe x gene)	aln	tree	aln+tree	aln	tree	aln+tree	aln	tree	aln+tree	aln	tree	aln+tree
treedepth = 0.5												
5x8	8	15	23	8	4	12	8	3	11	8	1	**9**
5x25	17	37	54	8	48	56	8	45	53	8	3	**11**
9x9	11	19	30	8	18	26	8	11	19	8	2	**10**
9x20	21	63	84	11	96	107	11	83	94	11	8	**19**
9x33	32	122	154	8	148	156	8	183	191	8	6	**14**
treedepth = 1												
5x8	8	15	23	2	8	10	2	5	7	2	2	**4**
5x25	16	44	60	2	75	77	2	60	62	2	11	**13**
9x9	11	20	31	1	21	22	1	14	15	1	3	**4**
9x20	21	65	86	5	98	103	5	116	121	5	8	**13**
9x33	26	98	124	2	217	219	2	216	218	2	15	**17**
treedepth = 1.5												
5x8	8	13	21	6	16	22	6	11	17	6	3	**9**
5x25	20	33	53	2	62	64	2	48	50	2	9	**11**
9x9	13	20	34	2	37	39	2	19	21	2	5	**7**
9x20	22	49	71	4	154	158	4	162	166	4	18	**22**
9x33	35	138	173	3	293	296	3	307	310	3	39	**42**

DOI:10.6814/NCCU202001688

Table 8. RF-distance analysis of the simulated data set (Best marked in Bold) where RAxMLv8.2.12 using 100 bootstraps, RAxML-ng using standard option, and IQ-TREE2 using standard option.

	BigBigTree				RAxML				RAxML-ng				IQ-TREE2			
set (spe x gene)	test1	test2	test3	ave	test1	test2	test3	ave	test1	test2	test3	ave	test1	test2	test3	ave
treedepth = 0.5																
5x8	18	30	32	**26.6**	30	22	28	**26.6**	32	24	30	28.6	34	24	30	29.3
5x25	134	124	128	**128.6**	152	140	174	155.3	150	144	172	155.3	158	154	174	162.0
9x9	74	60	34	**56.0**	70	82	64	72.0	84	92	84	86.6	78	84	78	80.0
9x20	226	214	242	**227.3**	222	228	236	228.6	230	248	242	240.0	246	252	262	253.3
9x33	524	466	464	484.6	474	474	480	**476.0**	484	502	504	496.6	500	496	504	500.0
treedepth = 1																
5x8	22	42	24	29.3	24	10	28	**20.6**	24	8	36	22.6	24	12	38	24.6
5x25	118	112	74	**101.3**	118	114	134	122.0	134	124	142	133.3	132	122	126	126.6
9x9	58	48	102	**69.3**	76	74	70	73.3	78	84	62	74.6	82	78	62	74.0
9x20	196	204	126	**175.3**	216	212	224	217.3	230	232	236	232.6	222	240	242	234.6
9x33	480	468	284	410.6	392	344	444	**393.3**	400	384	458	414.0	416	380	472	422.6
treedepth = 1.5																
5x8	46	20	10	25.3	12	38	14	**21.3**	18	40	18	25.3	18	42	16	25.3
5x25	96	72	104	**90.6**	136	160	130	142.0	138	164	136	146.0	152	170	146	156.0
9x9	40	44	44	**42.6**	64	74	60	66.0	66	76	72	71.3	70	82	74	75.3
9x20	152	100	158	**136.6**	218	228	212	219.3	218	256	210	228.0	224	248	218	230.0
9x33	492	400	372	421.3	322	352	406	**363.3**	328	378	432	379.0	336	366	422	374.6

3.2 Real biology data

The running time comparison is summarized in Table 9. We still divide the running time into three parts and analyze it, respectively. In the alignment part, the same results are obtained with simulation data. In the tree construction and combination parts, IQ-TREE2 also performs fastest with fast option. Besides, when the input sequences are in a small amount, BigBigTree has the same execution time with RAxML but slower than RAxML-ng. From microspore 232 and Olfactory receptor two datasets, we can find out that in the microspor dataset, BigBigTree has about 15 times faster than RAxML and 12 times faster than RAxML-ng. In the Olfactory receptor dataset, BigBigTree has about five times faster than RAxML and RAxML-ng. Therefore we know that the difference in speed rate has nothing to do with the total number of sequences. We believe it relates to the number of sequences in the cluster after BigBigTree clustered the sequences because the less amount of sequences are in the group after cluster, the less length of sequences will be after alignment, and the program can get better performance.

The log-likelihood comparison is summarized in Table 10. While BigBigTree gets similar performance with RAxML in Cladiomy dataset and better than others, it receives worse performance than three tools in the other two datasets. Note that the log-likelihood score is calculated with alignment sequences generated by MAFFT, which is also used as input to RAxML, RAxML-ng, and IQ-TREE2. Therefore, they will take advantage of log-likelihood.

DOI:10.6814/NCCU202001688

DOI:10.6814/NCCU202001688

Table 9. Running time (in secs) of the real data set.

	BigBigTree			RAxML			RAxML-ng			IQ-TREE2		
Data Set type	aln	tree	aln+tree	aln	tree	aln+tree	aln	tree	aln+tree	aln	tree	aln+tree
Cladiomy	2	10	12	2	7	9	2	2	4	2	1	3
microspor232	45	56	101	23	1524	1547	23	1250	1273	23	25	48
Olfactory receptor	93	1242	1335	10	6381	6391	10	6386	6396	10	85	95

Table 10. Log-likelihood analysis of the real data set.

Data Set	BigBigTree	RAxML	RAxML-ng	IQ-TREE2
Cladiomy	-8643.62	**-8643.61**	-8747.92	-8748.56
microspor	-162713.70	**-162028.51**	-162278.99	-162278.99
Olfactory receptor	-254403.86	-246185.27	**-246153.67**	-246153.70

4 Discussion

We get quite promising results in terms of running time against RAxML and RAxML-ng thanks to our divide and concatenate approach. According to the outcome of simulation biology data, we observe that BigBigTree performs well on RF distance in most cases, while the performance is significantly degraded on the largest dataset. We choose 9x33 with a treedepth 0.5 dataset for analysis, then two characteristics are found from the actual gene tree and clustering results. First, there are many in-paralogs. In-paralog is a paralog that was duplicated after the speciation and hence are orthologs to a cluster in the other species [32]. Second, the depth of the subtree is quite shallow (Figure 12). Both of these can cause clustering errors due to gene sequences are too similar.

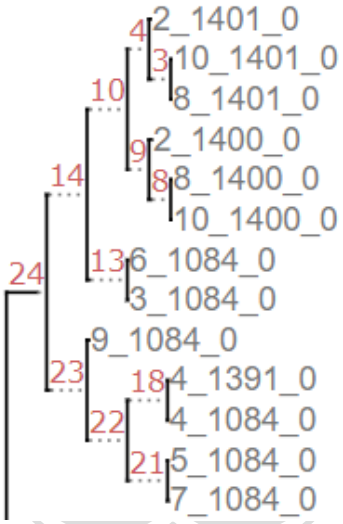


Figure 12：A subtree of 9x33 gene tree

In BigBigTree, the maximum size parameter is set to the number of species for a cluster. It may cause redundancy in concatenation when in-paralogs exist. Take a tree in Figure 13.a for example, if the maximum cluster size is set to 3, sequences may be divided into $\{a_1, b_1, c_1\}$, $\{a_2\}$ and $\{b_2, c_2, c_3\}$ (squares in Figure 13.b, m=3). Furthermore, the concatenate long-string alignments will be "a_1-b_1-c_1", "a_2-*gap*-*gap*" and "*gap*-b_2-c_2" (the up alignment in Figure 13.c) which contain a lot of noise from the gap. Otherwise, if the maximum cluster size is set to 4, we will get "a_1-b_1-c_1" and "a_2-b_2-c_2" (the bottom alignment in Figure 13.c), which gain more information than the previous one for tree construction.

DOI:10.6814/NCCU202001688

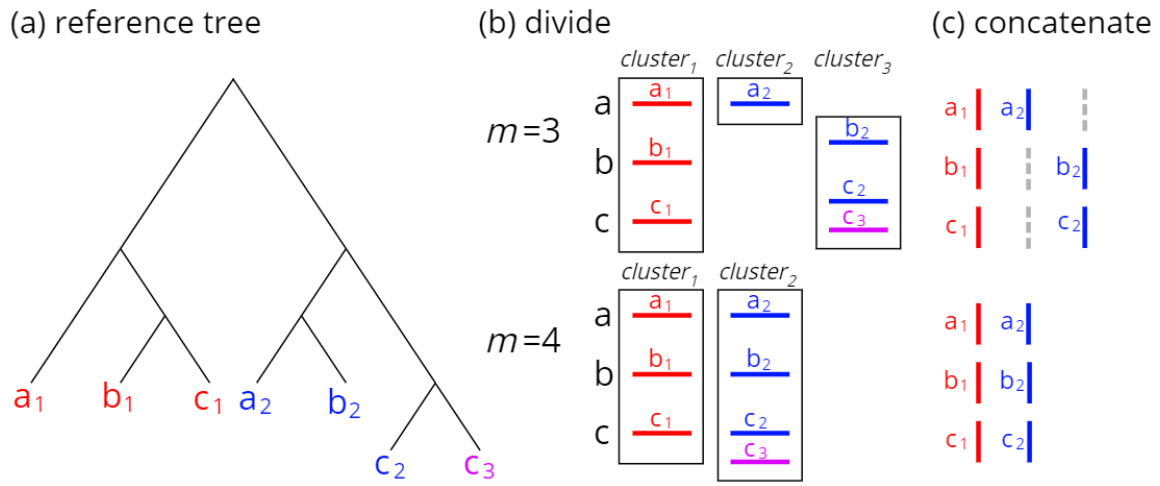


Figure 13：The influence of maximum cluster size, *m*, (a) a reference gene trees for three species, *a*, *b*, and *c* with two gene duplication events {1,2} and {2,3} (b) two clusters regarding *m*=3 and *m*=4, respectively (c) corresponding contacted long-string alignments based on (b).

Therefore, here we try to set the maximum size larger on 9x33 with a treedepth 0.5 dataset, and the results are summarized in Table 7. We set the maximum cluster number to 12, 15, and 18, respectively. After that, it almost comes out with better performance when the number increases. The results of test2 and test3 went backward when it increased to 18, but it is still better than 9. In recent years, many computational tools are created and developed for orthology analysis [33]. There is potential to replace *hcluster* with other clustering tools. *OrthoVenn* modified portions of the data processing steps in the OrthoMCL algorithm, replaced *BLAST* with *UBLAST,* and used *orthAgogue* to identify putative orthology and in-paralogy relations [34,35]. Moreover, *OrthoFinder* solves a previously undetected gene length bias in orthogroup inference, it does not adopt the pairwise strategy but instead attempts to identify complete orthogroups [36,37]. An orthogroup is the set of genes descended from a single gene in the last common ancestor of all the species being considered [38]. Applying these two methods for clustering may achieve better results than *hcluster*. Unfortunately, because of the difference in the execution environment and the requirements for the database, we have not been successfully put into the Nextflow framework for testing.

DOI:10.6814/NCCU202001688

Table 11. RF-distance analysis of the 9x33 datasets with treedepth=1 regarding difference maximum cluster size (best performance marked as bold).

Max. cluster size	test1	test2	test3	Ave.	Ave. time
9	524	466	464	484.6	154
12	496	438	450	461.3	146
15	462	**424**	**426**	437.3	124
18	**424**	444	428	**432.0**	116

We show the feasibility of BigBigTree; however, it still has some room to improve the method of ortholog clustering. It's just as well that the tool that meets the execution environment can be easily replaced in the Nextflow pipeline. Users can choose their favorite tools for clustering, alignment, and tree construction and replace it. With the advancement of sequencing technology and analysis tools in the future, BigBigTree can give improved results.

After brief visualizing tree topology, BigBigTree comes out with reasonable phylogenetics (Figure 14). We will further evaluate the quality of topology based on log-likelihood, investigate the limitation of our algorithm, and improve it through a more comprehensive benchmark.

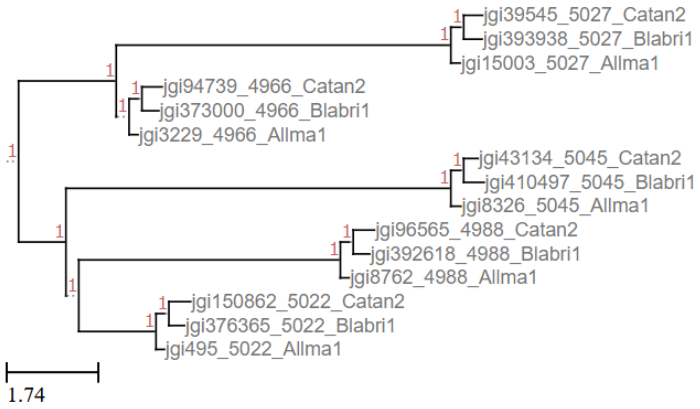


Figure 14：The snapshot of the Cladiomy tree by BigBigTree. TreeViewer draws the topology on ETE 3 [39].

DOI:10.6814/NCCU202001688

5 Conclusion

We not only implement a highly flexible and efficient Nextflow framework but also propose a novel approach, *BigBigTree*. The success of BigBigTree results from two factors: The Divide and Concatenate algorithm helps to reduce the time for tree construction and, thus, the concatenate long-string alignments enrich more information to complete phylogenetic tree accurately. BigBigTree has a great ability to build a large gene tree. It's necessary to increase the number of sequences with the development of next-generation sequencing and third-generation sequencing. After that, it is very promising to gain more knowledge from the large tree in the future.

DOI:10.6814/NCCU202001688

References

1. Contributors to Wikimedia projects. Phylogenetic tree. 2002 Nov 20 [cited 2020 May 19]; Available from: https://en.wikipedia.org/wiki/Phylogenetic_tree

2. BIL 106 - Lecture 4 [Internet]. [cited 2020 May 19]. Available from: http://www.bio.miami.edu/dana/106/106F05_4.html

3. Larget BR, Kotha SK, Dewey CN, Ané C. BUCKy: gene tree/species tree reconciliation with Bayesian concordance analysis. Bioinformatics. 2010 Nov 15;26(22):2910–1.

4. Mirarab S, Reaz R, Bayzid MS, Zimmermann T, Swenson MS, Warnow T. ASTRAL: genome-scale coalescent-based species tree estimation. Bioinformatics. 2014 Sep 1;30(17):i541–8.

5. Mirarab S, Warnow T. ASTRAL-II: coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. Bioinformatics. 2015 Jun 15;31(12):i44–52.

6. Vachaspati P, Warnow T. ASTRID: Accurate Species TRees from Internode Distances. BMC Genomics. 2015 Oct 2;16 Suppl 10:S3.

7. Lemoine F, -B. Domelevo Entfellner J, Wilkinson E, Correia D, Dávila Felipe M, De Oliveira T, et al. Renewing Felsenstein's phylogenetic bootstrap in the era of big data [Internet]. Vol. 556, Nature. 2018. p. 452–6. Available from: http://dx.doi.org/10.1038/s41586-018-0043-0

8. Delsuc F, Brinkmann H, Philippe H. Phylogenomics and the reconstruction of the tree of life [Internet]. Vol. 6, Nature Reviews Genetics. 2005. p. 361–75. Available from: http://dx.doi.org/10.1038/nrg1603

9. Rokas A, Williams BL, King N, Carroll SB. Genome-scale approaches to resolving incongruence in molecular phylogenies. Nature. 2003 Oct 23;425(6960):798–804.

10. Ashkenazy H, Sela I, Levy Karin E, Landan G, Pupko T. Multiple Sequence Alignment Averaging Improves Phylogeny Reconstruction. Syst Biol. 2019 Jan 1;68(1):117–30.

11. Chang J-M, Floden EW, Herrero J, Gascuel O, Di Tommaso P, Notredame C. Incorporating alignment uncertainty into Felsenstein's phylogenetic bootstrap to improve its reliability [Internet]. Bioinformatics. 2019. Available from: http://dx.doi.org/10.1093/bioinformatics/btz082

12. BLAST: Basic Local Alignment Search Tool [Internet]. [cited 2020 May 19]. Available from: https://blast.ncbi.nlm.nih.gov/Blast.cgi

13. hcluster [Internet]. PyPI. [cited 2020 May 19]. Available from: https://pypi.org/project/hcluster/

14. Notredame C, Higgins DG, Heringa J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol. 2000 Sep 8;302(1):205–17.

DOI:10.6814/NCCU202001688

15. Katoh K, Misawa K, Kuma K-I, Miyata T. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Res. 2002 Jul 15;30(14):3059–66.

16. TreeSoft: TreeBeST [Internet]. [cited 2020 May 19]. Available from: http://treesoft.sourceforge.net/treebest.shtml

17. Minh BQ, Schmidt HA, Chernomor O, Schrempf D, Woodhams MD, von Haeseler A, et al. IQ-TREE 2: New Models and Efficient Methods for Phylogenetic Inference in the Genomic Era. Mol Biol Evol. 2020 May 1;37(5):1530–4.

18. Nguyen L-T, Schmidt HA, von Haeseler A, Minh BQ. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Mol Biol Evol. 2015 Jan;32(1):268–74.

19. Guindon S, Dufayard J-F, Lefort V, Anisimova M, Hordijk W, Gascuel O. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. Syst Biol. 2010 May;59(3):307–21.

20. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nat Biotechnol. 2017 Apr 11;35(4):316–9.

21. Di Tommaso Jean-Francois Taly Javier Herrero Cedric Notredame J-MCMMP. A divide and concatenate strategy for the phylogenetic reconstruction of large orthologous datasets. SMBE poster. 2012;

22. Clustering Run - MCL Clusters - Microsporidia [Internet]. [cited 2020 May 19]. Available from: https://genome.jgi.doe.gov/clm/run/microsporidia-2017-01.1750;sjugmT?organismsGroup=microsporidia

23. Mallo D, De Oliveira Martins L, Posada D. SimPhy: Phylogenomic Simulation of Gene, Locus, and Species Trees. Syst Biol. 2016 Mar;65(2):334–44.

24. Fletcher W, Yang Z. INDELible: a flexible simulator of biological sequence evolution. Mol Biol Evol. 2009 Aug;26(8):1879–88.

25. Lafond M, Meghdari Miardan M, Sankoff D. Accurate prediction of orthologs in the presence of divergence after duplication. Bioinformatics. 2018 Jul 1;34(13):i366–75.

26. Robinson DF, Foulds LR. Comparison of phylogenetic trees [Internet]. Vol. 53, Mathematical Biosciences. 1981. p. 131–47. Available from: http://dx.doi.org/10.1016/0025-5564(81)90043-2

27. Cardona G, Llabrés M, Rosselló F, Valiente G. Metrics for phylogenetic networks I: generalizations of the Robinson-Foulds metric. IEEE/ACM Trans Comput Biol Bioinform. 2009 Jan;6(1):46–61.

28. Stamatakis A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics. 2014 May 1;30(9):1312–3.

29. Kozlov AM, Darriba D, Flouri T, Morel B, Stamatakis A. RAxML-NG: a fast, scalable and user-friendly tool for

DOI:10.6814/NCCU202001688

maximum likelihood phylogenetic inference. Bioinformatics. 2019 Nov 1;35(21):4453–5.

30. Jones DT, Taylor WR, Thornton JM. The rapid generation of mutation data matrices from protein sequences. Comput Appl Biosci. 1992 Jun;8(3):275–82.

31. Chang J-M, Di Tommaso P, Notredame C. TCS: a new multiple sequence alignment reliability measure to estimate alignment accuracy and improve phylogenetic tree reconstruction. Mol Biol Evol. 2014 Jun;31(6):1625–37.

32. Sonnhammer ELL, Koonin EV. Orthology, paralogy and proposed classification for paralog subtypes. Trends Genet. 2002 Dec;18(12):619–20.

33. Nichio BTL, Marchaukoski JN, Raittz RT. New Tools in Orthology Analysis: A Brief Review of Promising Perspectives. Front Genet. 2017 Oct 31;8:165.

34. Wang Y, Coleman-Derr D, Chen G, Gu YQ. OrthoVenn: a web server for genome wide comparison and annotation of orthologous clusters across multiple species. Nucleic Acids Res. 2015 Jul 1;43(W1):W78–84.

35. Xu L, Dong Z, Fang L, Luo Y, Wei Z, Guo H, et al. OrthoVenn2: a web server for whole-genome comparison and annotation of orthologous clusters across multiple species [Internet]. Vol. 47, Nucleic Acids Research. 2019. p. W52–8. Available from: http://dx.doi.org/10.1093/nar/gkz333

36. Emms DM, Kelly S. OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. Genome Biol. 2015 Aug 6;16:157.

37. Emms DM, Kelly S. OrthoFinder: phylogenetic orthology inference for comparative genomics. Genome Biol. 2019 Nov 14;20(1):238.

38. Altenhoff AM, Schneider A, Gonnet GH, Dessimoz C. OMA 2011: orthology inference among 1000 complete genomes. Nucleic Acids Res. 2011 Jan;39(Database issue):D289–94.

39. Huerta-Cepas J, Serra F, Bork P. ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. Mol Biol Evol. 2016 Jun;33(6):1635–8.

DOI:10.6814/NCCU202001688