Comparative Genomics 2018
Practical 4: Phylogenomics

Group 11
Fuqi Xu, Milda Valiukonyte, Shuhan Xu

Summary
In this practical, we practiced to the identification of orthologs in different genomes and studied the phylogenetic relationship between orthologs and genomes. To identify orthologs, we ran BLAST with one genome as the query and the other three genomes as the databases. We assumed the best hits in three databases for the same query sequence are the orthologs. And we selected ten ortholog clusters for phylogeny study. We aligned different orthologs and built phylogenic trees with different clusters using Belvu. To better analyze the evolutionary relationship, we also joined all ten orthologs clusters into a metagene and make the phylogenic tree. Bootstrap is used to test the confidence of the tree. To summarize the evolutionary relationship of different genomes based on phylogenetic analysis, we build a consensus tree with Phylip. The phylogenetic relationship of most orthologs agrees with both the metagene tree and the consensus tree, and they share same splits with the 16s rRNA tree, which indicates the conservation during evolution.

Exercise 1

4.1

Blast XML output for querying 09.fa.txt (reference) against 17.fa.txt:	see attachment 17_results
Blast XML output for querying 09.fa.txt (reference) against 49.fa.txt:	see attachment 49_results
Blast XML output for querying 09.fa.txt (reference) against 51.fa.txt:	see attachment 51_results

5.

Script for parsing XML output:	see attachment parser.py
Usage:	python3 parser.py <xml ouput file> > <parser output file>
Description:	the script parses through blast xml output file and prints out the tag for reference genome query protein id, query protein sequence, target genome best hit protein id and best hit protein sequence.

5.2

Parser output for 09.fa.txt against 17.fa.txt:	see attachment 17_parsed
Parser output for 09.fa.txt against 49.fa.txt:	see attachment 49_parsed
Parser output for 09.fa.txt against 51.fa.txt:	see attachmnt 51_parsed

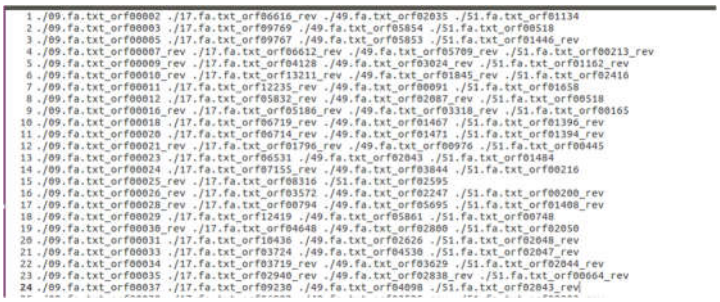
Screenshot for 17_parsed:

6.2

Script for clustering orthologs: see attachment cluster.py
Usage: python3 cluster.py
Description: the scripts reads ‘17_parsed’, ‘49_parsed’ and ‘51_parsed’ in the working directory and writes clusters of orthologs into the file ‘cluster’ in the same directory.

Cluster output: see attachment cluster

Screenshot for cluster:



7

Script for obtaining multi Fasta file: see attachment generate_fasta.py
Usage: python3 generate_fasta.py <cluster file> <cluster number> > <multi Fasta output file>
Description: For a specified cluster number in the input, the script looks up the cluster in the cluster file and retrieves the sequences from the proteome files. The headers and sequences are written to the multi Fasta ouput file.

7.1

Multi Fasta for cluster 1: see attachment multiFASTAcluster_1
Multi Fasta for cluster 2: see attachment multiFASTAcluster_2
Multi Fasta for cluster 3: see attachment multiFASTAcluster_3
Multi Fasta for cluster 4: see attachment multiFASTAcluster_4
Multi Fasta for cluster 5: see attachment multiFASTAcluster_5
Multi Fasta for cluster 6: see attachment multiFASTAcluster_6
Multi Fasta for cluster 7: see attachment multiFASTAcluster_7
Multi Fasta for cluster 8: see attachment multiFASTAcluster_8
Multi Fasta for cluster 9: see attachment multiFASTAcluster_9
Multi Fasta for cluster 10: see attachment multiFASTAcluster_10

7.2

Screenshot for multiFASTAcluster_1:

Exercise 2

1.

Alignment file for cluster 1: see attachment cluster_1_aligned
Alignment file for cluster 2: see attachment cluster_2_aligned
Alignment file for cluster 3: see attachment cluster_3_aligned
Alignment file for cluster 4: see attachment cluster_4_aligned
Alignment file for cluster 5: see attachment cluster_5_aligned
Alignment file for cluster 6: see attachment cluster_6_aligned
Alignment file for cluster 7: see attachment cluster_7_aligned
Alignment file for cluster 8: see attachment cluster_8_aligned
Alignment file for cluster 9: see attachment cluster_9_aligned
Alignment file for cluster 10: see attachment cluster_10_aligned

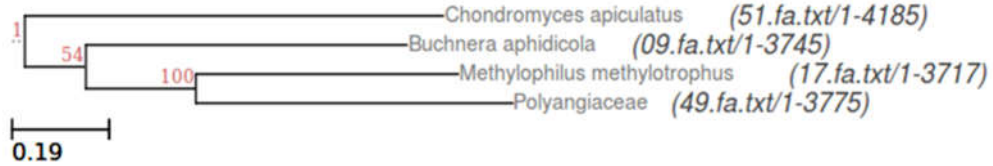
1.1

Script for concatenating alignments: see attachment combine.py
Usage: python3 combine.py
Description: The script reads the 10 cluster alignment files above and and concatenate them into one long alignment into the file ‘metagene_aligned’ in the working directory

File for the concatenated alignments: see attachment metagene_aligned.
Screenshot for metagene_aligned:

1.2

The tree was constructed using neighbor-joining with Scoredist distance correction



1.3
1000 bootstrap samples
<the same tree as 1.2 above>

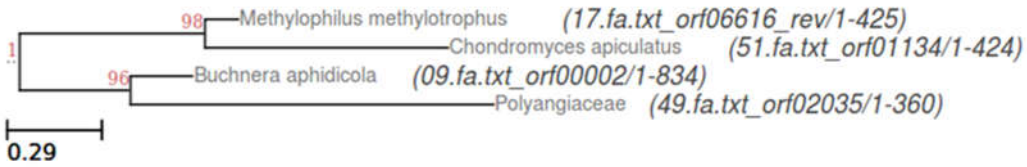
One of the split receive 100 percent boostrap support while the other split receives only 54 percent bootstrap support. The latter suggests that the tree is not able to adequately represent the evolutionary history of the metagene. This is because the metagene is composed of many genes, each of which evolves at different rate due to different evolutionary pressure experienced.

Exercise 3

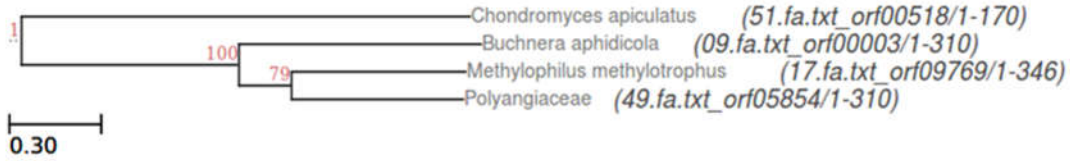
1.1
File for 10 different trees in Newick format: see attachment 10_trees_newick

1.2
The trees were constructed using Neighbor-joining with Scoredist distance correction with 1000 bootstrap samples each

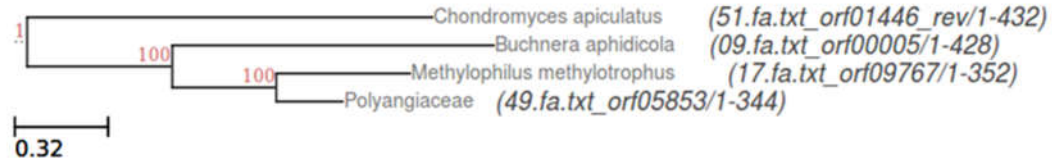
Cluster 1



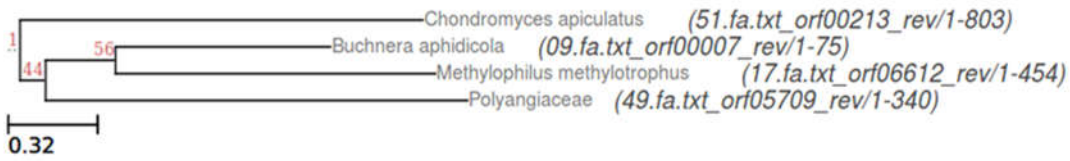
Cluster 2



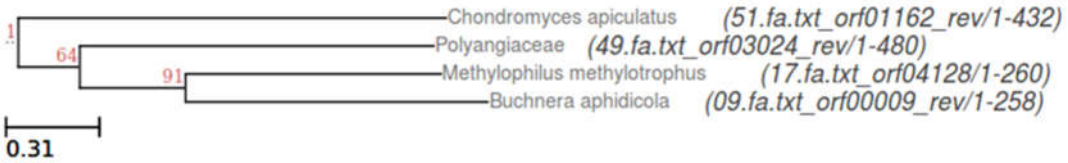
Cluster 3



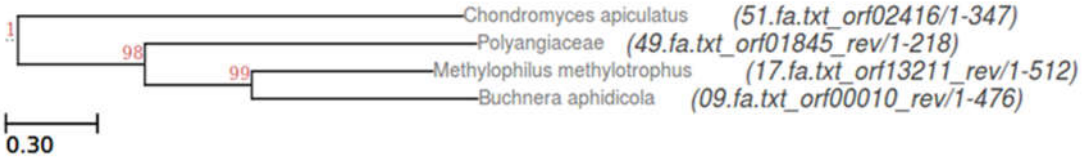
Cluster 4



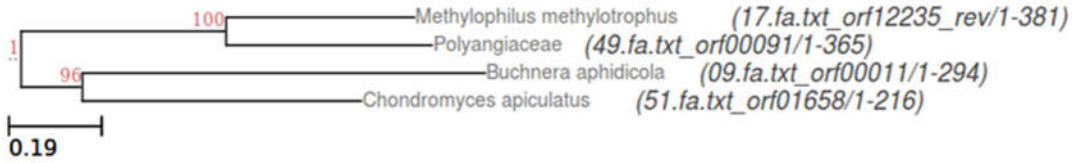
Cluster 5



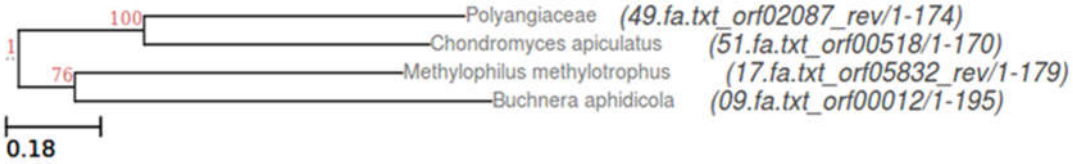
Cluster 6



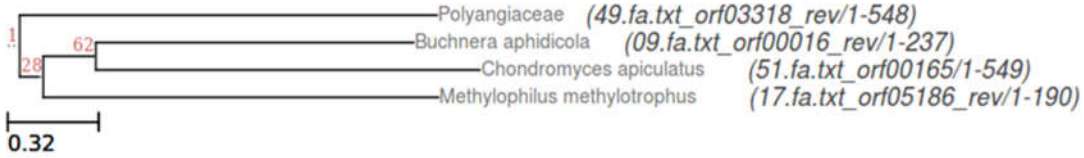
Cluster 7



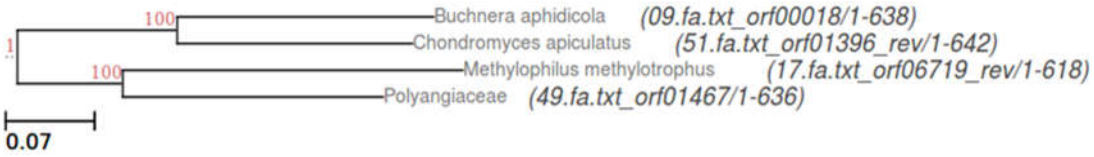
Cluster 8



Cluster 9



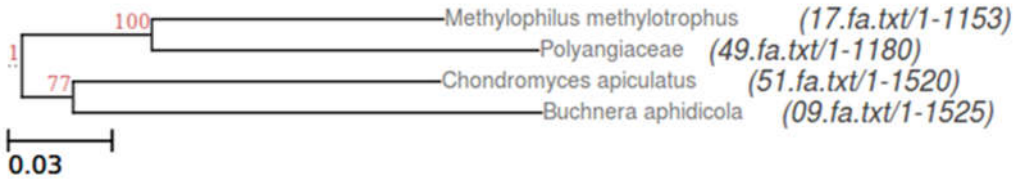
Cluster 10



Some trees receives approximately 100 percent bootstrap support for both splits (clusters 1, 3, 6, 7, 10), suggesting that the data show strong support for the topologies when neighbor-joining and scoredist distance corrections are used. On the other hand, there are a few trees with very poor bootstrap support (cluster 4 and 9 with less than 50 percent support for one of the split). This could be due to incorrect gene prediction from glimmer or inaccurate multiple sequence alignment. Also, the sequences in cluster 4 and cluster 9 are very short. The length of the sequence also influence the confidence of the tree.

Among the tress that have good bootstrap values, the topologies of the trees may be different (such as cluster 1 and 3). This may be due to the different selection pressure and hence different evolutionary rate of the different genes in the same organism.

1.3
We assume that the species tree follows the topology of the gene tree for 16S rRNA from practical 3. See the tree below.



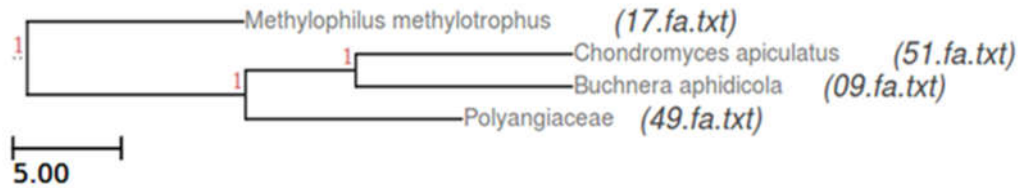
The topology of the metagene tree is the same as 16S rRNA tree (although the branch lengths are different), suggesting that the evolution of most genes in the metagene follows the evolution of the species. In particular, the topologies of clusters 2, 3, 7, 9 and 10 are the same as the topology of 16S rRNA tree.

On the other hand, cluster 1 (bifunctional aspartate kinase/homoserine dehydrogenase I) and cluster 6 (sodium:alanine symporter family protein) have different topologies from the those of clusters 2, 3, 7, 9 and 10 and 16S rRNA.

1.4
Cluster 1 (bifunctional aspartate kinase/homoserine dehydrogenase I) and cluster 6 (sodium:alanine symporter family protein) have different topologies from the other clusters because the bifunctional enzyme and symporter experience different evolutionary pressure from the other genes in the same organism. In particular, the bifunctional aspartate kinase/homoserine dehydrogenase I is involved in biomolecule synthesis while the sodium:alanine symporter is involved in biomolecule transport. Both genes could be tightly linked to the presence of nutrients in the environment and hence experience a possibly higher evolutionary pressure compared to the other genes.

2.1.2
intree: see attachment intree

2.1.3
Consensus tree



The consensus tree has the same topology as the meta-tree.

Phylip consense program looks for common monophyletic groups among the input trees and represents them in the consensus tree. If majority rule consensus type is selected in the terminal, groups which occurs in at least 50 percent of the input trees will be represented in the consensus tree. If strict consensus type is chosen, only groups which appears in all the input trees will be retained in the consensus tree. If Majority Rule (extended) is selected, groups which occurs in less than 50 percent of the input trees will also be used in consensus tree building as long as they do not conflict

with the more frequent groups. The M_l option allows the user to specify the fraction of occurrence in the input trees for the groups to be represented in the consensus tree.

The program also asks if the tree is to be treated as rooted and which species is the outgroup (the default is the first species on the first tree.