Lucas De Vrieze (r0665032)

Introduction

In comparative genomics, information from whole-genome sequences is compared to extract and predict genomic features for many different organisms. These genomic feature may include genes, regulatory regions, intron-exon structures etc. With the current deluge of newly available genomic information, many new discoveries can be made and new genomic mechanisms discovered. Comparing the genomic information of two randomly chosen organisms can already yield useful insights about their mutual ancestry and their respective evolutionary paths. This assignment did so for two seemingly different species: the **Gloeobacter violaceus** (GV), a cyanobacterium, and **Theobroma cacao** (TC), the cacao plant.

1 Methods

Orthologs, paralogs and co-orthologs identification

The proteomes of GV and TC were downloaded from NCBI Genome (respective RefSeq entries: GCF_000011385.1 (4452 proteins) and GCF_000208745.1 (30854 proteins)). For each entry, a BLAST database was constructed using the `makeblastdb` command of the BLAST+ 2.7.1 module. Co-orthologs are groups of paralogs in one species of which one protein is orthologous to a protein of another species. To find co-orthologs, both orthologs and paralogs need to be identified.
The orthologs for these two species were sought using best bidirectional hits (BBH) by BLASTing their respective proteomes against each other. The paralogs, at the other hand, were determined via BLASTing the proteomes against themselves and applying a significance threshold of 0.00001 on the output. This work was carried out by submitting the `blast_both_all.pbs` script to the VSC and running the `bbh.py` Python script on the output files.
This script also finds proteins in TC that are co-orthologous with a protein in GV. Therefore, it collects TC proteins that point to the same GV protein as their best hit, on condition that one of these TC proteins is the BBH of the GV protein, all TC proteins meet a hit significance threshold of 0.00001 and all have a paralog in TC. To limit the data volume, I filtered for groups with no more than five proteins. From the resulting collection, I randomly picked a GV protein and its TC co-orthologs.

Phylogenetic tree construction

BLASTing the two BBH proteins against the database of non-redundant proteins at NCBI would yield two clusters of proteins around each one of the queried proteins. In order to get a more evenly distributed sample of all evolutionary lineages, I handpicked 25 species along a variety of taxonomic lineages by submitting the GV protein at STRING and picking species from the tree in the gene co-occurrence window.
Then, I BLASTed a fasta file with the sequences of the GV protein and its two TC co-orthologs against the proteomes of these 25 species and, again, applied a significance threshold of 0.00001 on the output. These 25 proteomes were obtained in the same way as the ones of GV and TC and are listed below in Table 1. All 25 species got homologs assigned, of which the sequences were collected in one fasta file. The `get_homologs.pbs` HPC script carried out the BLASTing and collected the protein IDs, from which Python script `homologs.py` created the fasta file.
This fasta file was fed to ClustalO 1.2.4 [1] at default settings to generate a multiple sequence alignment, which was at its turn fed to ClustalW 2.1 [2] in phylogenetic tree mode at default settings. As such, a phylogenetic tree was constructed using neighbour-joining and 10000 bootstraps, and uploaded to iTol [3] for visualisation.
A species tree was constructed from the complete proteomes of the 25 species using OrthoFinder 2.5.4 [4] in default settings, and uploaded to iTOl as well.

Species	***RefSeq entry***
Pseudomonas aeruginosa	GCF_000006765.1
Escherichia coli	GCF_000005845.2
Klebsiella pneumoniae	GCF_000240185.1
Enterobacter cloacae	GCF_023702375.1
Nitrosomonas eutropha	GCF_000014765.1
Rhodobacter capsulatus	GCF_000021865.1
Nitrospira defluvii	GCF_905220995.1
Streptomyces coelicolor	GCF_000203835.1
Clostridium perfringens	GCF_020138775.1
Lactobacillus rhamnosus	GCF_006151905.1
Staphylococcus aureus	GCF_000013425.1
Listeria monocytogenes	GCF_000196035.1
Bacillus subtilis	GCF_000009045.1
Cyanobium gracile	GCF_000316515.1
Gloeobacter violaceus	GCF_000011385.1
Theobroma cacao	GCF_000208745.1
Helianthus annuus	GCF_002127325.2
Chlamydomonas reinhardtii	GCF_000002595.2
Micromonas pusilla	GCF_000151265.2
Fusarium oxysporum	GCF_000271745.1
Saccharomyces cerevisiae	GCF_000146045.2
Yarrowia lipolytica	GCF_000002525.2
Hydra vulgaris	GCF_022113875.1
Caenorhabditis elegans	GCF_000002985.6
Drosophila melanogaster	GCF_000001215.4

Table 1: RefSeq entries for the 25 handpicked species for which protein datasets were downloaded.

Conserved regions

Conserved regions were identified by scanning the MSA using a simplified version of the trident conservation score metric of Valdar [5], smoothed by a moving average with a window size of 5. This approach allows to take gaps in the alignment into account as well. The conservation score of a column x of symbols was defined as

$$C(x) = (1 - SE(x)) \cdot (1 - G(x)) \tag{1}$$

$SE(x)$ is the Shannon entropy scaled to a range between 0 and 1 with p_i the empirical probability of observing symbol i and K the number of symbols (21: the 20 amino acids and the gap symbol):

$$SE(x) = \lambda \sum_{i}^{K} p_i \log_2 p_i \tag{2}$$

The scaling factor λ is defined as following, with N the number of sequences in the MSA:

$$\lambda = [\log_2(\min(N, K))]^{-1} \tag{3}$$

$G(x)$ is the gappiness or simply the fraction of gap symbols in column x.
The Python script `conservation.py` calculates this conservation metric for my sequence alignment.

2 Results & discussion

BBHs and homology

The protein datasets of GV and TC contain respectively 4452 and 30854 protein coding genes. Respectively 4392 and 30599 got a best hit assigned, from which 1991 orthologs have been identified via the BBHs.
In GV, there were 15517 paralogous hits, while in TC, there were 1706580. Applying the criteria defined above, 410 co-orthologous groups were extracted from these hits. I randomly picked one from these: the group co-orthologous with GV protein WP_011140090.1, or the 50S ribosomal protein L4. This co-orthologous group contains two TC proteins, XP_007026182.2 and XP_017984011.1, of which the second one is the BBH. XP_007026182.2 is the 50S ribosomal protein L4 of TC, while XP_017984011.1 is the 50S ribosomal protein L4 of TC **chloroplasts**. Remarkably, the GV protein has a stronger hit with the chloroplastic protein than with the cytoplasmatic one.

Phylogenetic tree

After handpicking 25 species and getting their 29 homologs to the proteins of the co-orthologous group, the phylogenetic tree in Figure 1 was generated from the aligned homologous sequences. The species tree generated by OrthoFinder from the 25 proteomes is depicted in Figure 2.

A first observation that stands out by comparing the homology tree and the species tree, is the good agreement in groups. I added a similar colouring for the labels of species that are consistently grouped in both trees. Mostly, there is only one significant hit matching with the chosen co-orthologous group for the 25 species. In case there are multiple, it mostly concerns differently localised proteins, e.g. cytoplasmatic ribosomal proteins vs. chloroplastic ones. Therefore, there are barely no duplication nodes, only speciation nodes, which is no surprise given the single-copy character of ribosomal proteins. Only for *Helianthus annuus*, there were three hits: one chloroplastic ribosomal, one cytoplasmatic ribosomal and an uncharacterised protein. Yet, the latter contains an ribosomal L4/L1-like domain, although for both L4 and L1 proteins, cytoplasmatic as well as chloroplastic proteins have been annotated. Perhaps, it is the mitochondrial variant, as sunflowers have both chloroplasts and mitochondria cell organelles.
Furthermore, this phylogenetic tree shows some evidence for the endosymbiotic theory. Chloroplastic proteins are grouped together with proteins of cyanobacteria and unicellular photosynthetic eukaryotes, while mitochondrial ones are related to proteins of heterothrophic prokaryotes. This is clear for the case of fungi, but less so for the multicellular clade, which falls completely in the brownish coloured group. This group contains both mitochondrial and cytoplasmatic proteins, interspersed with an uncharacterised *Chlamydomonas* protein and the *Rhodobacter* protein. In the branching pattern from the ancestral node next to the *Chlamydomonas* protein branch, I would hypothesise that the upper branch is mitochondrial, while the lower one is cytoplasmatic. This would constitute a similar structure as in the fungi case. Nevertheless, the bootstrap support for this branching pattern is not that strong, especially not at the deeper nodes.

To conclude, the picked co-orthologous group is not really an example of co-orthology, as the apparent paralogs are the result of independent speciation events rather than duplications. I would consider this a special case of pseudoparalogy, in which an entire genome would have been horizontally 'transferred' to a new cell organelle due to the endosymbiosis. As such, two orthologs, i.e. the host's and the endosymbiont's, might have been united in the total gene pool of one organism.

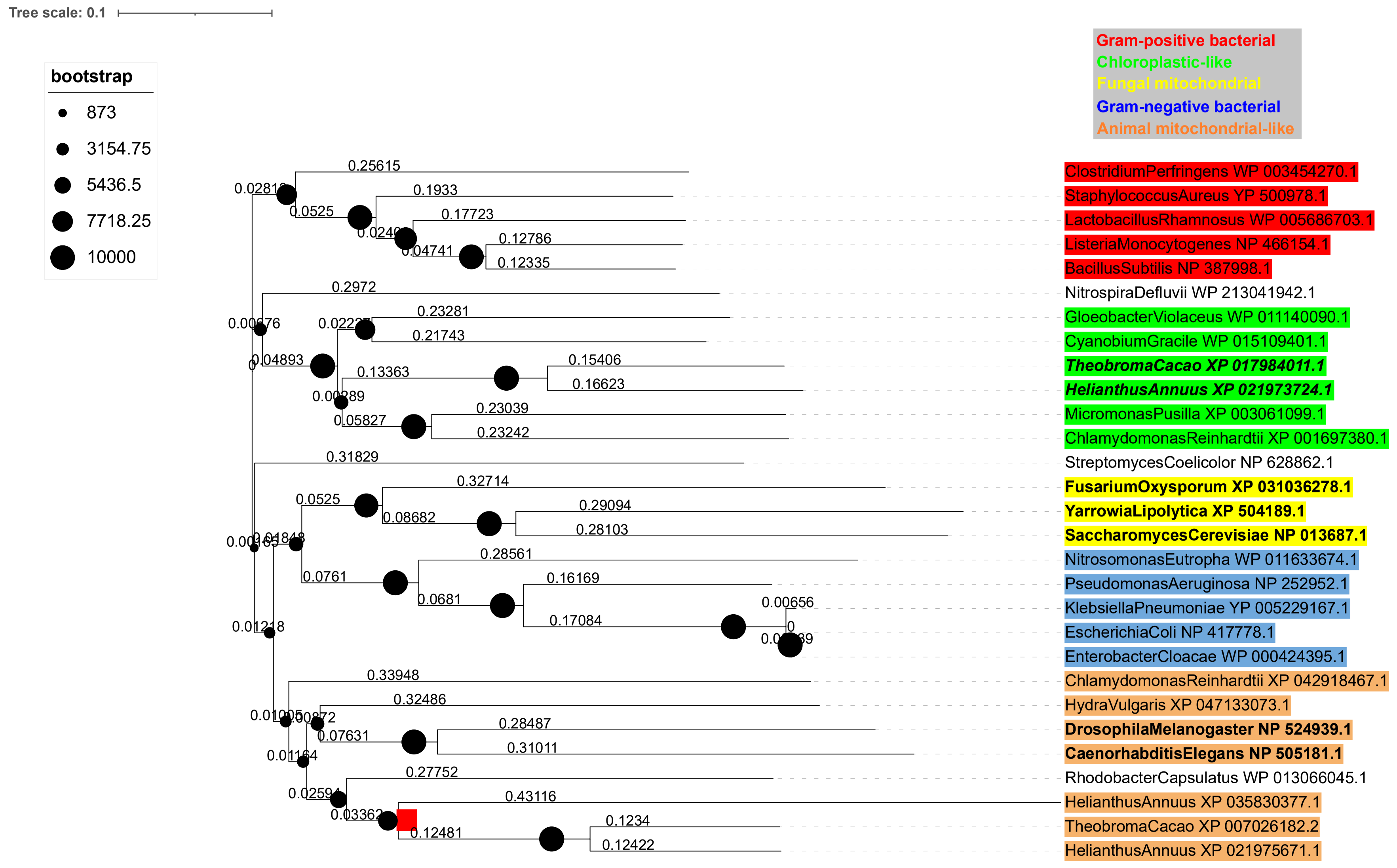
Tree scale: 0.1
bootstrap
873
3154.75
5436.5
7718.25
10000
Gram-positive bacterial
Chloroplastic-like
Fungal mitochondrial
Gram-negative bacterial
Animal mitochondrial-like
ClostridiumPerfringens WP 003454270.1
StaphylococcusAureus YP 500978.1
LactobacillusRhamnosus WP 005686703.1
ListeriaMonocytogenes NP 466154.1
BacillusSubtilis NP 387998.1
NitrospiraDefluvii WP 213041942.1
GloeobacterViolaceus WP 011140090.1
CyanobiumGracile WP 015109401.1
TheobromaCacao XP 017984011.1
HelianthusAnnuus XP 021973724.1
MicromonasPusilla XP 003061099.1
ChlamydomonasReinhardtii XP 001697380.1
StreptomycesCoelicolor NP 628862.1
FusariumOxysporum XP 031036278.1
YarrowiaLipolytica XP 504189.1
SaccharomycesCerevisiae NP 013687.1
NitrosomonasEutropha WP 011633674.1
PseudomonasAeruginosa NP 252952.1
KlebsiellaPneumoniae YP 005229167.1
EscherichiaColi NP 417778.1
EnterobacterCloacae WP 000424395.1
ChlamydomonasReinhardtii XP 042918467.1
HydraVulgaris XP 047133073.1
DrosophilaMelanogaster NP 524939.1
CaenorhabditisElegans NP 505181.1
RhodobacterCapsulatus WP 013066045.1
HelianthusAnnuus XP 035830377.1
TheobromaCacao XP 007026182.2
HelianthusAnnuus XP 021975671.1
0.25615
0.1933
0.17723
0.12786
0.12335
0.2972
0.23281
0.21743
0.15406
0.16623
0.13363
0.23039
0.23242
0.31829
0.32714
0.29094
0.28103
0.28561
0.16169
0.17084
0.00656
0.33948
0.32486
0.28487
0.31011
0.27752
0.43116
0.1234
0.12422
0.12481

Figure 1: Neighbour-joining phylogenetic tree with 10000 bootstraps (circular labels) and branch lengths. A red square indicates the apparent duplication node. The labels are coloured by consistently grouped lineages (e.g. Gram-positive bacterial, chloroplastic-like). Labels in bold are mitochondrial proteins, while labels in bold italics are chloroplastic ones. Proteins for which no localisation annotation was given, are assumed to be cytoplasmatic.

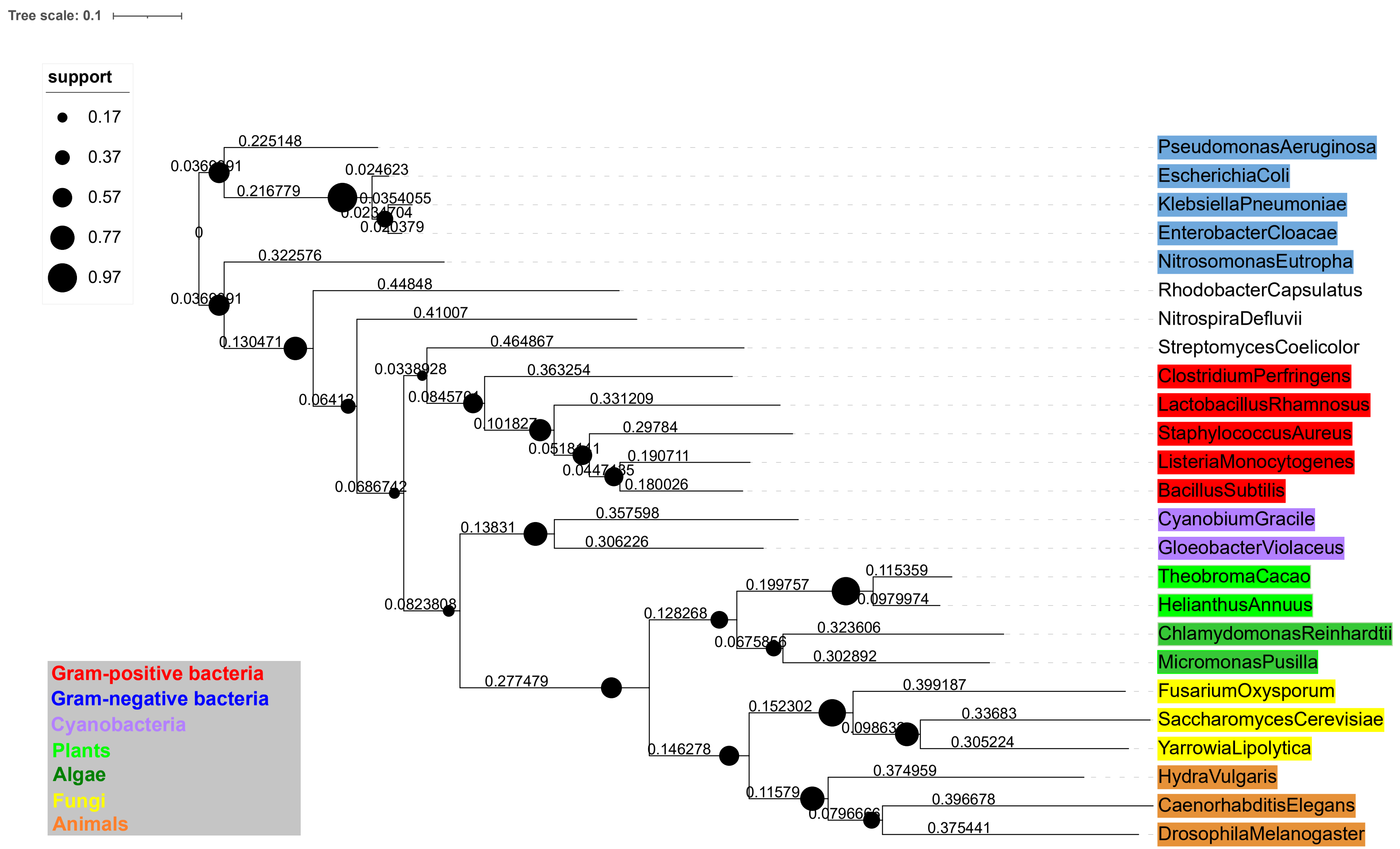

Figure 2: OrthoFinder species tree with support values (circular labels) and branch lengths. The labels are coloured by consistent groups of species (e.g. Gram-positive bacteria, plants, animals).

Conservation

The conservation of all sequences in the MSA was assessed using the dual score metric outlined in section 1, which resulted in Figure 3 below. Setting an ad hoc threshold at 0.3, three conserved stretches were identified.

- a large stretch from residues 180 to 320
- a very small one of about 6 residues around residue 343
- a relatively large one from residues 395 to 460

All homologs contain a ribosomal L4 domain according to the batch CD-Search tool of NCBI [6]. Two domain hits were reported: both the bacterial type (TIGR03953)[1] and the eukaryotic type (pfam00573)[2]. Both are part of the ribosomal protein L4/L1 family (cl00325)[3].
Exploring the alignment of the seed sequences of these domains shows roughly the same conserved structures: a large stretch, a very short one of about 5 to 7 residues and a relatively large one.

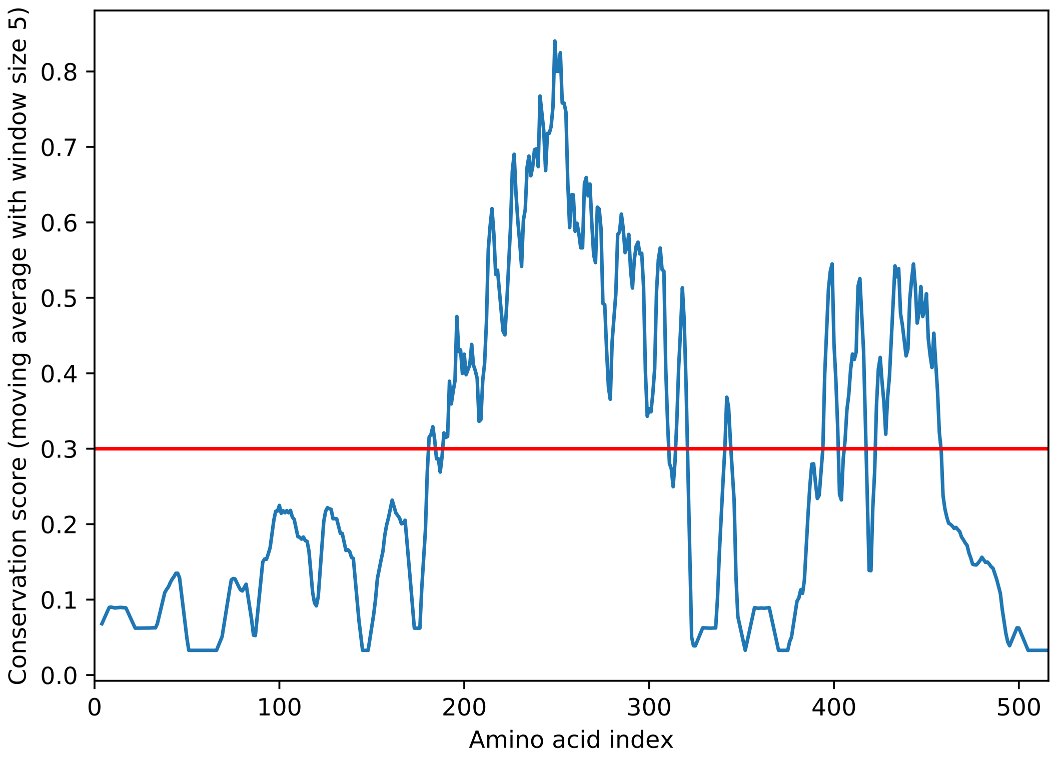

Figure 3: Conservation score along the length of the aligned sequences.

[1]https://www.ncbi.nlm.nih.gov/Structure/cdd/cddsrv.cgi?uid=274877
[2]https://www.ncbi.nlm.nih.gov/Structure/cdd/cddsrv.cgi?uid=425758
[3]https://www.ncbi.nlm.nih.gov/Structure/cdd/cddsrv.cgi?uid=444837

References

[1] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, *et al.*, "Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega," *Molecular systems biology*, vol. 7, no. 1, p. 539, 2011.

[2] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, *et al.*, "Clustal w and clustal x version 2.0," *bioinformatics*, vol. 23, no. 21, pp. 2947–2948, 2007.

[3] I. Letunic and P. Bork, "Interactive tree of life (itol) v5: an online tool for phylogenetic tree display and annotation," *Nucleic acids research*, vol. 49, no. W1, pp. W293–W296, 2021.

[4] D. M. Emms and S. Kelly, "Orthofinder: phylogenetic orthology inference for comparative genomics," *Genome biology*, vol. 20, no. 1, pp. 1–14, 2019.

[5] W. S. Valdar, "Scoring residue conservation," *Proteins: structure, function, and bioinformatics*, vol. 48, no. 2, pp. 227–241, 2002.

[6] S. Lu, J. Wang, F. Chitsaz, M. K. Derbyshire, R. C. Geer, N. R. Gonzales, M. Gwadz, D. I. Hurwitz, G. H. Marchler, J. S. Song, *et al.*, "Cdd/sparcle: the conserved domain database in 2020," *Nucleic acids research*, vol. 48, no. D1, pp. D265–D268, 2020.

Appendices

Listing 1: blast_both_all.pbs

```
#!/bin/bash
#PBS -A lp_edu_evol_quant_genetics
#PBS -l walltime=1:00:00
#PBS -l nodes=1:ppn=8

cd $VSC_SCRATCH/CompAss
module purge
module load BLAST+

makeblastdb -in GloeobacterViolaceus.faa -dbtype prot -parse_seqids -out
   GV
makeblastdb -in TheobromaCacao.faa -dbtype prot -parse_seqids -out TC
blastp -query GloeobacterViolaceus.faa -db TC -outfmt 6 -out GVqTCd_a.out
    -num_threads 8
blastp -query TheobromaCacao.faa -db GV -outfmt 6 -out TCqGVd_a.out -
   num_threads 8
blastp -query GloeobacterViolaceus.faa -db GV -outfmt 6 -out GVqGVd_a.out
    -num_threads 8
blastp -query TheobromaCacao.faa -db TC -outfmt 6 -out TCqTCd_a.out -
   num_threads 8
```

Listing 2: bbh.py

```
#!/usr/bin/env python3
"""
@author: Lucas De Vrieze (r0665032)
Comparative genomics - graded assignment

Orthology, paralogy and co-orthology
"""

# See PSB file blast_both_all.pbs for bash commands to generate the
    reciprocal blast results files.

import csv
from collections import Counter

## Reading result files
with open("TCqGVd_a.out", "r") as TCGV_handle:
    TCGV_reader = csv.reader(TCGV_handle, delimiter="\t")
    TCGV = [entry for line in TCGV_reader for entry in line]
with open("GVqTCd_a.out", "r") as GVTC_handle:
    GVTC_reader = csv.reader(GVTC_handle, delimiter="\t")
    GVTC = [entry for line in GVTC_reader for entry in line]
with open("GVqGVd_a.out", "r") as GVGV_handle:
    GVGV_reader = csv.reader(GVGV_handle, delimiter="\t")
    GVGV = [entry for line in GVGV_reader for entry in line]
with open("TCqTCd_a.out", "r") as TCTC_handle:
    TCTC_reader = csv.reader(TCTC_handle, delimiter="\t")
    TCTC = [entry for line in TCTC_reader for entry in line]

## Find orthologs via BBHs
# getting lists of best hits for each protein of both organisms from
   BLAST output
TC_besthits = dict(zip(reversed(TCGV[0::12]), reversed(TCGV[1::12])))
GV_besthits = dict(zip(reversed(GVTC[0::12]), reversed(GVTC[1::12])))
# combine to find BBHs
orthologs = {}
for h in GV_besthits.keys():
    try:
        if TC_besthits[GV_besthits[h]] == h:
            orthologs[h] = GV_besthits[h]
    except:
        continue

# nr. of protein coding genes: count number of entries in Fasta files
# grep -c '>' GloeobacterViolaceus.faa -> 4452
# grep -c '>' TheobromaCacao.faa -> 30854
# Some proteins didn't generate sufficiently strong hits (GV: 4392 best
    hits; TC: 30599 best hits)
# nr. of orthologs = nr. of BBHs -> 1991
```

```
## Find paralogs via significant self-BLASTing hits
# Getting lists of hit self-pairs with probabilities
TCTC_probs = dict(zip(list(zip(TCTC[0::12], TCTC[1::12])), TCTC[10::12]))
GVGV_probs = dict(zip(list(zip(GVGV[0::12], GVGV[1::12])), GVGV[10::12]))
# Extracting significant self-pairs to get paralogs
TC_paralogs = [pair for pair,prob in TCTC_probs.items() if float(prob) <=
    0.00001 and pair[0] != pair[1]]
TC_prots_with_paralogs = list(set([pair[0] for pair in TC_paralogs]))
GV_paralogs = [pair for pair,prob in GVGV_probs.items() if float(prob) <=
    0.00001 and pair[0] != pair[1]]
GV_prots_with_paralogs = list(set([pair[0] for pair in GV_paralogs]))

## Proteins in TC homologous with a GV protein
# Getting all TC-GV pairs with probabilities from BLAST output
TCGV_probs = dict(zip(list(zip(TCGV[0::12], TCGV[1::12])), TCGV[10::12]))
GVTC_probs = dict(zip(list(zip(GVTC[0::12], GVTC[1::12])), GVTC[10::12]))

# Selecting groups of TC proteins significantly co-orthologous with a GV
   protein, that have a BBH and at least one paralog
# Filtering for groups not larger than 5
counts_all_coorthologs = dict(Counter(TC_besthits.values()))
coorthologs = {}
for k in counts_all_coorthologs.keys():
    if k in orthologs.keys():
        group = [pair for pair,prob in TCGV_probs.items() if k == pair[1]
            and pair[0] in TC_prots_with_paralogs and float(prob) <=
           0.00001]
        if len(group) > 1 and len(group) <= 5:
            coorthologs[k] = group

# Just picking one from the list: WP_011140090.1
# Find the TC proteins that point to WP_011140090.1 as their best hit
GV_prot = "WP_011140090.1"
TC_prot = [p[0] for p in coorthologs[GV_prot]]
prot_ortholog = orthologs["WP_011140090.1"]
print(GV_prot)
print("Co-orthologs:\t" + str(TC_prot))
print("BBH:\t" + prot_ortholog)

# Writing orthologs results file
with open("orthologs.txt", "w") as res_file:
    for GV,TC in orthologs.items():
        res_file.write(GV + "\t" + TC + "\n")

# Writing paralogs results file
with open("paralogs.txt", "w") as res_file:
    for pair in TC_paralogs:
        res_file.write(pair[0] + "\t" + pair[1] + "\n")
    for pair in GV_paralogs:
        res_file.write(pair[0] + "\t" + pair[1] + "\n")
```

```
# Writing co-orthologs results file
with open("coorthologs.txt", "w") as res_file:
    for pairs in coorthologs.values():
        for pair in pairs:
            res_file.write(pair[0] + "\t" + pair[1] + "\n")
```

Listing 3: get_homologs.pbs

```
#!/bin/bash
#PBS -A lp_edu_evol_quant_genetics
#PBS -l walltime=0:05:00
#PBS -l nodes=1:ppn=8

cd $VSC_SCRATCH/CompAss/trees
module purge
module load BLAST+

list=$(dir proteomes)
for entry in $list
do
makeblastdb -in ./proteomes/$entry -dbtype prot -parse-seqids -out $entry
blastp -query bbh.fasta -db $entry -outfmt 6 -out ./blast/$entry.out -
   num_threads 8
done

cd blast
rm -f res.filtered
list=$(dir)
for file in $list
do
cat $file | awk '$11␣<␣0.00001' >> res.filtered
done
cat res.filtered | awk '{print␣$2}' | sort -u > homologs_ids
```

Listing 4: homologs.py

```
#!/usr/bin/env python3
"""
@author: Lucas De Vrieze (r0665032)
Comparative genomics - graded assignment

Getting sequences of homologs in 25 handpicked proteomes
"""

# See PSB file get_homologs.pbs for bash commands to detect the homologs
    in all 25 species by blast'ing

from Bio import SeqIO
import os

with open("homologs_ids", "r") as id_handle:
    ids = id_handle.read().split("\n")[:-1]

# Scan all proteome fasta files for the entries to get, make a user-
    friendly label and export them to a new fasta file
proteomes = os.listdir('proteomes')
records = []
for fasta in proteomes:
    proteome = SeqIO.to_dict(SeqIO.parse('/'.join(["proteomes",fasta]), "
        fasta"))
    for ID in ids:
        if ID in proteome.keys():
            orig = proteome[ID]
            new_id = fasta.split('.')[0] + ':' + ID
            # Keeping out a partial sequence without start codon
            if orig.seq[0] != 'M':
                continue
            record = SeqIO.SeqRecord(orig.seq, id = new_id, name = new_id
                , description = orig.description)
            records.append(record)
with open("homologs.fasta", "w") as handle:
    SeqIO.write(records, handle, "fasta")
```

Listing 5: conservation.py

```
#!/usr/bin/env python3
"""
@author: Lucas De Vrieze (r0665032)
Comparative genomics − graded assignment

Conservation score metrics of the alignment of homologs from 25 species
"""

# MSA was generated using clustalO at default settings.

from Bio import AlignIO
from collections import Counter
from numpy import log2
import matplotlib.pyplot as plt
from pandas import Series

# Read alignment file
with open("homologs.msa", "r") as align_handle:
    alignment_reader = AlignIO.read(align_handle, "fasta")
    alignment = [str(entry.seq) for entry in alignment_reader]

# Calculate the conservation score
## Two−prong approach inspired by the trident score of Valdar (https://
    doi.org/10.1002/prot.10146)
scores = []
nr_alignments = len(alignment)
for i in range(len(alignment[0])):
    a_i = [a[i] for a in alignment]

    # Shannon entropy
    aa_counts = dict(Counter(a_i))
    fractions = [v/nr_alignments for v in aa_counts.values()]
    scale_factor = 1/(log2(min(nr_alignments, 21)))
    shannon = −scale_factor*sum([p*log2(p) for p in fractions])

    # Gappiness
    try:
        gappiness = aa_counts['−']/nr_alignments
    except:
        gappiness = 0

    # Combined score
    score = (1−shannon)*(1−gappiness)
    scores.append(score)

# setting up a moving average for plotting
window_size = 5
scores_series = Series(scores)
windows = scores_series.rolling(window_size)
```

```
moving_average = windows.mean().tolist()

# plotting
plt.figure()
plt.plot(moving_average)
plt.axhline(y=0.3, color='red')
plt.xlabel('Amino␣acid␣index')
plt.ylabel('Conservation␣score␣(moving␣average␣with␣window␣size␣5)')
plt.xlim([0,len(alignment[0])])
plt.tight_layout()
plt.show()
plt.savefig("ConservationScore.pdf")
```