Prof. Dr. Daniel Huson
Algorithmen der Bioinformatik
Fachbereich Informatik
Mathematisch-Naturwissenschaftliche Fakultät

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Sequence Bioinformatics **WS 2021/22**

Assignment 8 **Due: 22-Dec-2021, 10 am**

The goal of this assignment is to implement a simple version of Mash, based on the lecture notes and on the original paper (`https://doi.org/10.1186/s13059-016-0997-x`). Your program should read as input a "command", the k-mer size k, a sketch size s and then, finally, a list of input fastA files.

mash.py ⟨command⟩ -k ⟨k-mer size⟩ -s ⟨sketch size⟩ ⟨input-files⟩

Possible commands are: *sketch*, *jaccard* and *distances*.

1 Implementation of sketching (4 points)

When the command *sketch* is specified, for each of the input files, the program should compute a bottom sketch of size s and report all sketches to the console.

2 Implementation of Jaccard index (3 points)

When the command *jaccard* is specified, for each pair of input files, the program should compute the Jaccard index, and report the indices to the console. (in the format below).

3 Implementation of Mash distance (1 point)

When the command *distances* is specified, for each pair of input files, the program should compute the Mash distance, and report the distances to the console (in the format below).

4 Bacterial tree (2 points)

Download the file `genomes.zip` from Ilias. Compute Mash distances between all input files, using **k=17** and **s=800**.

Use a program such as SplitsTree4, or a web-resource, to compute the neighbor-joining tree for the distances.

Example of format for Jaccard indices and distances (for a subset of 4 of the genomes):

```
4
Candidatus_Accumulibacter_aalborgensis_3 0.00000000 0.21218974 0.23848626 0.21772365
Candidatus_Accumulibacter_phosphatis_Bin19 0.21218974 0.00000000 0.22384866 0.20714391
Candidatus_Accumulibacter_phosphatis_HKU-1 0.23848626 0.22384866 0.00000000 0.21218974
Candidatus_Accumulibacter_phosphatis_UBA2327 0.21772365 0.20714391 0.21218974 0.00000000
```

Some hints:

Structure your program as follows:

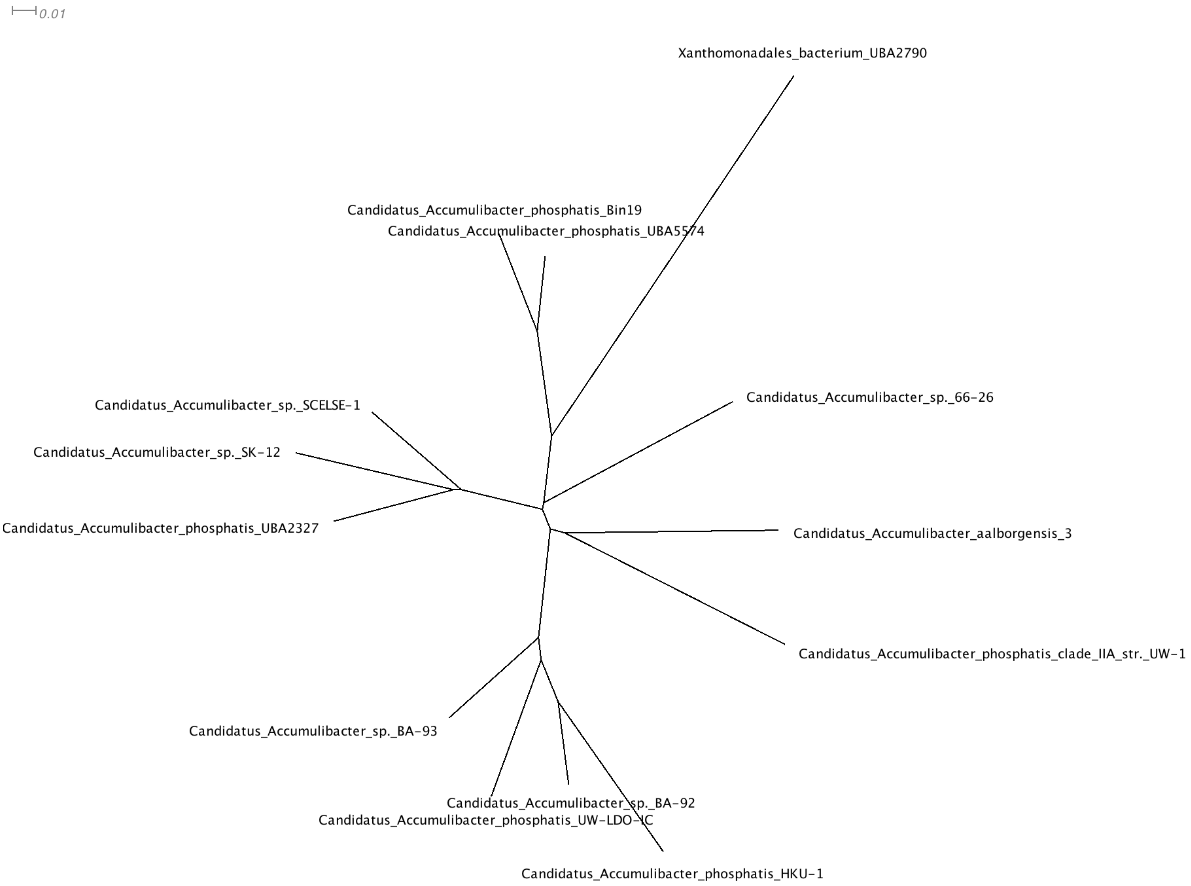
1. parse the command-line options to get:
    - the command
    - k and s
    - all the input files
2. For each input file: compute the sketch
3. if *command* is "sketch": output all sketches, as name of file and then one hash-value per line, in descending order
4. if *command* is "jaccard" or "distances":

```
for each pair of input files:
    compute the Jaccard index
    if command is "jaccard":
        store the Jaccard index
    else:
        compute the Mash distance from the Jaccard index
        store the distance
Print the stored values (Jaccard indices or Mash distances)
```

The example of distances shown at the bottom of the previous page are Mash distances produced by my implementation of the method. (There were computed with k=17 and s=800).

Please *do not* implement the use of a minimum coverage threshold c.

The resulting tree should look something like this:


0.01
Xanthomonadales_bacterium_UBA2790
Candidatus_Accumulibacter_phosphatis_Bin19
Candidatus_Accumulibacter_phosphatis_UBA5574
Candidatus_Accumulibacter_sp._SCELSE-1
Candidatus_Accumulibacter_sp._SK-12
Candidatus_Accumulibacter_phosphatis_UBA2327
Candidatus_Accumulibacter_sp._66-26
Candidatus_Accumulibacter_aalborgensis_3
Candidatus_Accumulibacter_phosphatis_clade_IIA_str._UW-1
Candidatus_Accumulibacter_sp._BA-93
Candidatus_Accumulibacter_sp._BA-92
Candidatus_Accumulibacter_phosphatis_UW-LDO-IC
Candidatus_Accumulibacter_phosphatis_HKU-1

So, some of the genomes are very similar, whereas one is very different from all others.