**ANI of SAR86 Genomes**

[November 02, 2020](https://nicolemicrobiome.blogspot.com/2020/11/ani-of-sar86-genomes.html)

 Goal: perform an all to all pairwise Average Nucleotide Identity (ANI) comparison of COMPLETE SAR86 genomes.

1. Isolate genomes with completeness exceeding 50% (as calculated by Pachiadaki 2019).

I used the PATRIC metadata of our SAR86 genomes and merged it with Table S2 from Pachiadaki 2019 (gorg-tropics\_sags\_tables2). This was done in R: I imported both datasets, removed the first 68 rows from the patric table (leaving only gorg-tropics samples), removed all columns except for "SAG" and "Genome Completeness" from the S2 table. The "Strain" column in patric is equivalent to the "SAG" column in S2, so I merged the two on that column, keeping all the patric rows but cutting any unmatched S2 rows (essentially a left outer join).

Now that each of our patric genomes are matched up the their completeness... I filtered the new dataset on completeness >= 50. This resulted in 412 SAR86 genomes.

2. Download fastANI.

$ wget https://github.com/ParBLiSS/FastANI/releases/download/v1.32/fastANI-Linux64-v1.32.zip

$ unzip fastANI-Linux64-v1.32.zip

$ module load gcc/9.3.0

$ ./fastANI

3. Setup fastANI input files.

For a many vs many comparison, fastANI requires two files containing the paths to all the genomes you wish to compare (a "query" file and a "reference" file, but in my case these will be identical files).

Since I already had all the genome IDs in a spreadsheet in R, it was trivial to tack on the path to each ID within R.

"/u/home/n/nzeltser/project-ngarud/sar86/"+GENOME\_ID+".fna"

4. Run ANI

$ ./fastANI --ql project-ngarud/fastANI\_input\_genomes.txt --rl project-ngarud/fastANI\_input\_genomes.txt -o /u/scratch/n/nzeltser/SAR86/fastani.out

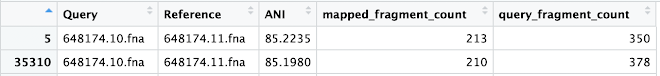
5. Data processing.

For an all vs all comparison, even if many combinations fall below the ANI threshold and are not displayed, there will be a lot of duplicates...

First of all, pairwise comparisons to each other:

[https://1.bp.blogspot.com/-yBQ_nEkrFA0/X6BSTOefqNI/AAAAAAAAkP0/-G-H4OfoQUs7yuG_jPkcpjdg1kOOI9duACLcBGAsYHQ/w644-h51/Screen%2BShot%2B2020-11-02%2Bat%2B10.38.38%2BAM.png](https://1.bp.blogspot.com/-yBQ_nEkrFA0/X6BSTOefqNI/AAAAAAAAkP0/-G-H4OfoQUs7yuG_jPkcpjdg1kOOI9duACLcBGAsYHQ/s1302/Screen+Shot+2020-11-02+at+10.38.38+AM.png)

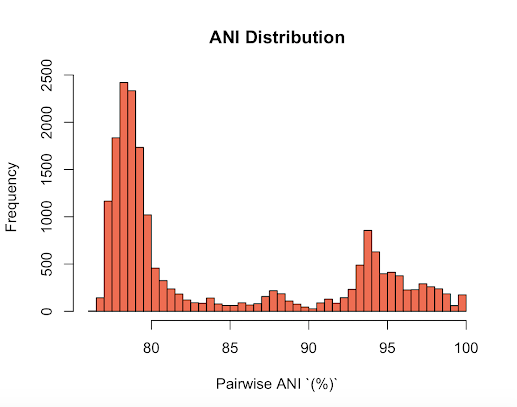
Second of all, when Query and Reference are flipped. In this example one of the pairings was flipped back so that they could be shown together as duplicates:

[](https://1.bp.blogspot.com/-ihkYVEKEKzA/X6BTCyBaG9I/AAAAAAAAkP8/COu6PXcodv0EUBgoO0JTurQgBOs3Wm9zACLcBGAsYHQ/s1282/Screen+Shot+2020-11-02+at+10.41.38+AM.png)

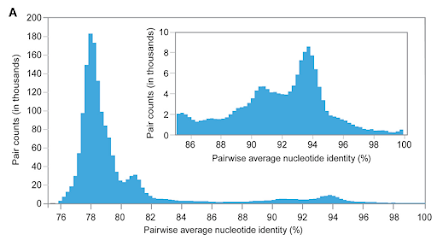
Even though the same pair is being compared, since originally the "query" and "reference" designation was reversed for one of the pairs, the computed results are different (ANI is different). Which ANI result should be chosen to represent each pair? Does it matter?

When presenting a distribution of ANI for the population, is it necessary to remove these duplicate pairs, or should they be treated as individual results?

This distribution includes ANI for all pairs except for self pairs (SAR86 genomes from gorg-tropics).

[](https://1.bp.blogspot.com/-sE3heSbf444/X6BXkilCv9I/AAAAAAAAkQI/nurB7NwaCwcSvQEe3cI_APRlxVHgzA_jgCLcBGAsYHQ/s1110/Screen+Shot+2020-11-02+at+11.00.21+AM.png)

Compare to Pachiadaki (the sample size here is much larger, this one includes many more species):

[](https://1.bp.blogspot.com/-P6-kkesgTaE/X6BX2rUSgyI/AAAAAAAAkQQ/7vbXG9NAgQEgxUBJlYn_ziZCpbvlMjv3wCLcBGAsYHQ/s1162/Screen+Shot+2020-11-02+at+10.58.24+AM.png)

**ANI Dendrogram using Bacsort Pipeline**

[November 23, 2020](https://nicolemicrobiome.blogspot.com/2020/11/ani-dendrogram-using-bacsort-pipeline.html)

 So a little while ago we discovered this great step-by-step guide to... fixing RefSeq assembly labeling?

Well that's the main purpose of the pipeline but within it are the very relevant steps of

1. Performing a pairwise ANI analysis

2. Constructing a distance matrix

3. Creating a dendrogram

<https://github.com/rrwick/Bacsort>

Previously, I had used the matrix file outputted by fastANI as an input to various clustering algorithms in the Phylip toolkit.

Instead, Bacsort includes its own custom script for creating a distance matrix from the pairwise ANI output.

$ pairwise\_identities\_to\_distance\_matrix.py --max\_dist 0.2

The max\_dist option sets the maximum allowed genomic distance, 0.2 representing 80% ANI. This means that any pairwise comparison that diverged more than 80% is going to be coerced to 80%. Essentially, whereas fastANI outputs, this script will substitute "0.2"

The distances are calculated with the function (1-ANI)/100     hence 0.2 = 80% ANI

I also handles seeing the same pair twice (each will appear a second time, but in reverse order).

This distance matrix is then assembled into a Newick tree using a neighbor-joining algorithm provided through an R package.

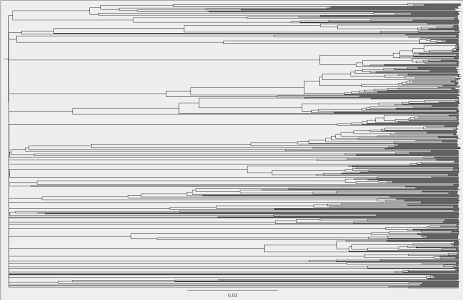
The clustering algorithm is described by this paper: <https://academic.oup.com/mbe/article/14/7/685/1119804>

The R packages involved are ape (providing the bionj clustering) and phangorn (providing a function for reading in a phylip formatted distance matrix).

A custom R script is provided:

bionj\_tree.R

The output from this script is a Newick tree, which I can display as previously, using FigTree:

[](https://1.bp.blogspot.com/-btf0E0iCoMo/X7yRHVXHPdI/AAAAAAAAkmw/KdLdOOA7Ihg8pADXtSOGf1dnMuMcX7-nACLcBGAsYHQ/s2048/Screen+Shot+2020-11-23+at+8.49.52+PM.png)

This is a much more decipherable tree, especially due to the lack of negative values.

Additionally, it is faithful to the pairwise ANI output. Samples that you would expect to be closely related based on ANI are actually close together in the tree.

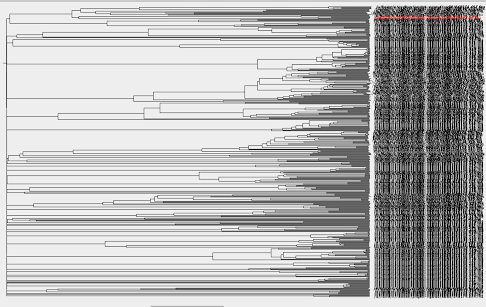
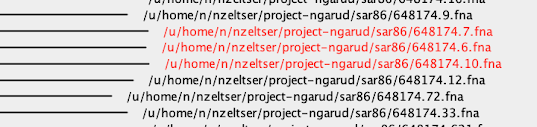
Repeating the examples from the previous post...

Sample 10 vs Sample 6 ANI > 97%

Sample 10 vs Sample 7 ANI > 97%

Sample 7 vs Sample 6 ANI > 97%

Each of these samples is colored in red in the following image (zoomed in in the second image):

[](https://1.bp.blogspot.com/-5GE6FBh2Mto/X7ySXLgGBQI/AAAAAAAAkm8/nECWAZhy6Lc393auTwWh1PzctUd-Qt8uACLcBGAsYHQ/s2048/Screen+Shot+2020-11-23+at+8.55.31+PM.png)[](https://1.bp.blogspot.com/-5MIeaSHZkXU/X7ySdwIrFrI/AAAAAAAAknA/qS79IdLFiuQDxlx5KNJOqInMu89MV9uRACLcBGAsYHQ/s720/Screen+Shot+2020-11-23+at+8.55.03+PM.png)

These samples are now right next to each other, instead of being all over the map, as they were in my previous attempt at a tree.

Let's check with MUMMER:

Sample 624 vs Sample 440

-> Result, very good alignment.

                               [REF]                [QRY]

[Sequences]

TotalSeqs                         31                   27

AlignedSeqs             30(96.7742%)         24(88.8889%)

UnalignedSeqs             1(3.2258%)          3(11.1111%)

[Bases]

TotalBases                    835045               925880

AlignedBases        654841(78.4198%)     654857(70.7281%)

UnalignedBases      180204(21.5802%)     271023(29.2719%)

Sample 566 vs Sample 381

-> Result: very good alignment

                               [REF]                [QRY]

[Sequences]

TotalSeqs                         34                    5

AlignedSeqs             34(100.0000%)          5(100.0000%)

UnalignedSeqs             0(0.0000%)           0(0.0000%)

[Bases]

TotalBases                    994772              1284767

AlignedBases        978807(98.3951%)     978803(76.1853%)

UnalignedBases        15965(1.6049%)     305964(23.8147%)

Sample 60 vs 107

-> Result: good alignment

                               [REF]                [QRY]

[Sequences]

TotalSeqs                         55                   29

AlignedSeqs             41(74.5455%)         25(86.2069%)

UnalignedSeqs           14(25.4545%)          4(13.7931%)

[Bases]

TotalBases                    979287              1143113

AlignedBases        665778(67.9860%)     665765(58.2414%)

UnalignedBases      313509(32.0140%)     477348(41.7586%)

Let's look at samples from distant clades:

Sample 6 vs Sample 107

-> Result: terrible alignment, as expected.

                               [REF]                [QRY]

[Sequences]

TotalSeqs                         43                   29

AlignedSeqs               4(9.3023%)          4(13.7931%)

UnalignedSeqs           39(90.6977%)         25(86.2069%)

[Bases]

TotalBases                   1250184              1143113

AlignedBases           3578(0.2862%)        3598(0.3148%)

UnalignedBases     1246606(99.7138%)    1139515(99.6852%)

Next steps:

We can try using mummer for pairwise alignment, using the pairwise ANI results as a guide. Keep in mind that mummer can only compare two sequences at a time.

and/or

We can use progressiveCactus (if I manage to get it running) to perform a multiple sequence alignment, guided by the tree that I just made.

Look into the alignment outputs from these two tools -> how can these data be processed/interpreted afterwards?