Infant Gut Microbiome Project Notebook

Elizabeth McDaniel

2022-03-31

Contents

Transposons in the Infant Gut Microbiome

[summary of the project, sections]

Initial Contigs Inspection

First I inspected the contigs to look at the data I was dealing with:

```
head 0118A1023.contigs.fa
```

```
>qb307082014_I_scaffold_1012 id=7301749 bin="0118A1023_UNK"
CCCGGCGTACGGGGCTGGGATTTGACGTCAACCGATTGCATCTTATCTTGGCTGTATTGG
AAAAACGCCTGCGCCTGAACTTCGGTCAGGTCGACATTTATGCCAAGGTCGGCGGCGGCA
TGAAGATTCAGGAGCCGGGCATGGACCTCGCGCTCGTGGCGGCGATGTTGTCGTCCTTCT
ACGACGTGCCGCTTCCCGAGCGGGCCGTGCTGTGGGGCGAAGTGGACCTCAACGGTCAGA
TCCGCCCCGTGGCCGCGCACGATATCCGGCTTTCGCAGGCGCGCAGGCTTGGCTACAAAC
CGATCCTTTTTCCTTCGCAGGGCGAGGGCGACGGAATCGCCACGGTCGTGGAGTTGCAGG
ACAGGCTGTTCCGCCGCAAGAACTGACGGCGGAAGCGGAAAGGGCCTCAAGTCGGGCGGG
CGCGGATCTCCACGGGGCCGGTTTTGGCGTGTGGGGAAAAAATGCATGACGCTCCCCTGT
TGCTCTTGCGCGGCCCTTGCGAACGCGTTATCAGGAATTATTCTTTATTCAAAAGGACGA
```

Since it looked like this was a file that had already been assembled into genome bins, I wanted to know how many genomes were within this assembly and how many scaffolds each genome had:

```
grep '>' 0118A1023.contigs.fa | awk -F " " '{print $3}' | sort | uniq -c
```

Which produced:

```
 157 bin="0118A1023_Bacteroides_vulgatus-like_42_520"
  23 bin="0118A1023_Citrobacter_koseri_53_430"
 922 bin="0118A1023_Clostridium_difficile_28_5"
  93 bin="0118A1023_Enterococcus_faecalis_36_53"
   5 bin="0118A1023_Enterococcus_faecalis_phage"
   1 bin="0118A1023_Klebsiella_pneumoniae_plasmid"
   7 bin="0118A1023_Proteobacteria_phage"
 104 bin="0118A1023_UNK"
```

It appears that there are 8 genomes in total, 4 of which are bacteria, 2 are phage, 1 a plasmid, and 1 unknown genome. To confirm this and possibly get a better idea of what the "unknown" genome is, I profiled the contigs with Anvi'o.

Anvi'o doesn't like long contig names, so I first created a plain file connecting each scaffold name to each bin name:

`grep '>' 0118A1023.contigs.fa | awk -F " " '{print $1"\t"$3}' > infant-gut.stb`, which looks like:

```
>qb307082014_I_scaffold_1012    bin="0118A1023_UNK"
>qb307082014_I_scaffold_1021    bin="0118A1023_UNK"
>qb307082014_I_scaffold_1039    bin="0118A1023_UNK"
>qb307082014_I_scaffold_1068    bin="0118A1023_UNK"
>qb307082014_I_scaffold_1069    bin="0118A1023_UNK"
>qb307082014_I_scaffold_1074    bin="0118A1023_UNK"
>qb307082014_I_scaffold_1094    bin="0118A1023_UNK"
>qb307082014_I_scaffold_1098    bin="0118A1023_UNK"
>qb307082014_I_scaffold_1104    bin="0118A1023_UNK"
>qb307082014_I_scaffold_1110    bin="0118A1023_UNK"
```

Then I shortened the contig names with `sed 's/\s.*$//' 0118A1023.contigs.fa > infant-gut-contigs.fa` so now the contig names look like:

```
>qb307082014_I_scaffold_1012
>qb307082014_I_scaffold_1021
>qb307082014_I_scaffold_1039
>qb307082014_I_scaffold_1068
>qb307082014_I_scaffold_1069
>qb307082014_I_scaffold_1074
>qb307082014_I_scaffold_1094
>qb307082014_I_scaffold_1098
>qb307082014_I_scaffold_1104
>qb307082014_I_scaffold_1110
```

I then generated a contigs database in Anvi'o with `anvi-gen-contigs-database -f infant-gut-contigs.fa -o contigs.db -n INFANTGUT -T 12`. This will identify ORFs in contigs with prodigal, which produced:

`Result .....: Prodigal (v2.6.3) has identified 17152 genes.` in ~2 minutes.

I then wanted to know the distribution of single copy core genes among these genomes, and what their taxonomy is. First I ran `anvi-run-hmms -c contigs.db --num-threads 12` which identifies sets of HMMs in the pre-made databases for `Ribosomal_RNA_16S, Ribosomal_RNA_5S, Ribosomal_RNA_28S, Protista_83, Ribosomal_RNA_12S, Archaea_76, Bacteria_71, Ribosomal_RNA_18S, Ribosomal_RNA_23S`. This pipeline can also use HMMs made from other sources, but the `Bacteria_71` and ribosomal sets will suffice for inspecting the contigs. This took less than 1 minute with 12 threads on a compute cluster.

To then assess the taxonomy of these single copy core genes in the metagenome, I first ran `anvi-run-scg-taxonomy -c contigs.db --num-parallel-processes 12 --num-threads 12 --min-percent-identity 90 --all-hits-output-file scg-taxonomy.txt`. This will return all taxonomy classifications to the single copy core genes with at least 90% identity based on the Genome Taxonomy Database (GTDB). This took less than 30 seconds with 12 threads. This output is in `results/scg-taxonomy.txt` which gives any taxonomical result matching the given single copy core gene with above 90% identity.

I then used this information to assess the taxonomical composition of the entire metagenome based on the classification of a single copy core gene using: `anvi-estimate-scg-taxonomy -c contigs.db -T 12 --metagenome-mode -o infant-gut-tax.txt` which will use one of the single copy core genes (ribosomal L2 in this case), which the output looks like (the file is also viewable in `results/infant-gut-tax.txt`):

```
scg_name    percent_identity    t_domain    t_phylum    t_class t_order t_family    t_genus t_species
Ribosomal_L2_3113   99.6    Bacteria    Firmicutes  Clostridia  Peptostreptococcales    Peptostreptococ
Ribosomal_L2_14412  98.5    Bacteria    Proteobacteria  Gammaproteobacteria Enterobacterales    Enteroba
Ribosomal_L2_792    98.9    Bacteria    Bacteroidota    Bacteroidia Bacteroidales   Bacteroidaceae  Pho
Ribosomal_L2_8091   99.6    Bacteria    Firmicutes  Bacilli Lactobacillales Enterococcaceae Enterococcus
Ribosomal_L2_5822   99.1    Bacteria    Desulfobacterota    Desulfovibrionia    Desulfovibrionales  Desu
```

From this check, there are 5 Ribosomal L2 single copy genes in the metagenome, and these are the classifications. To ensure that these classifications match up with the classifications given for the bins in the initial scaffolds file, we can export the gene calls and HMM hits from Anvi'o with the scripts:

```
anvi-export-gene-calls -c contigs.db --gene-caller prodigal -o gene_calls.txt anvi-script-get-hmm-hits-p
-c contigs.db -o hmm-hits.txt --hmm-source Bacteria_71
```

I then used these files to import into R for some basic stats and comparing the taxonomy of the Ribosomal L2 gene to the given taxonomy in the contigs file.

I imported the gene calls and corresponding scaffolds to bins into R, and joined these tables to create a table of all genes for all corresponding bins:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# import gene calls, scaffold names, and corresponding scaffolds to bins
gene_calls <- read.table("results/gene_calls.txt", header=TRUE, sep="\t") %>%
  select(gene_callers_id, contig)

scaffold_to_bin <- read.table("results/infant-gut.stb", col.names = c("contig", "bin"), sep="\t") %>%
  mutate(contig = gsub(">", "", contig))

# full table of gene accessions for each bin
genes_table <- left_join(gene_calls, scaffold_to_bin)
```

```
## Joining, by = "contig"
```

```
head(genes_table)
```

```
##   gene_callers_id                      contig               bin
## 1               0 qb307082014_I_scaffold_1012 bin=0118A1023_UNK
## 2               1 qb307082014_I_scaffold_1012 bin=0118A1023_UNK
## 3               2 qb307082014_I_scaffold_1012 bin=0118A1023_UNK
## 4               3 qb307082014_I_scaffold_1117 bin=0118A1023_UNK
## 5               4 qb307082014_I_scaffold_1168 bin=0118A1023_UNK
## 6               5 qb307082014_I_scaffold_1168 bin=0118A1023_UNK
```

I then imported all HMM hits for the Bacteria_71 set to both confirm the classifications for the Ribosomal L2 gene and do some statistics on how many single copy core genes were found in the contigs for each bin:

```
# HMM hits for Bacteria_71 collection to look at Ribosomal L2 hits and aggregating by genome to assess
hmm_hits <- read.table("results/hmm-hits.txt", header=TRUE, sep="\t") %>%
```

```
  select(gene_callers_id, source, gene_name, gene_hmm_id)

# join the genes table with the HMM hits to correspond with bin information
hmm_table <- left_join(hmm_hits, genes_table)
```

```
## Joining, by = "gene_callers_id"
```

```
hmm_table %>%
  select(gene_name) %>%
  unique() %>%
  count()
```

```
##    n
## 1 71
```

```
hmm_table %>%
  group_by(bin) %>%
  unique() %>%
  count()
```

```
## # A tibble: 5 x 2
## # Groups:   bin [5]
##   bin                                                 n
##   <chr>                                           <int>
## 1 bin=0118A1023_Bacteroides_vulgatus-like_42_520    73
## 2 bin=0118A1023_Citrobacter_koseri_53_430           71
## 3 bin=0118A1023_Clostridium_difficile_28_5          75
## 4 bin=0118A1023_Enterococcus_faecalis_36_53         74
## 5 bin=0118A1023_UNK                                  4
```

There are 71 unique single copy core genes that were identified among these contigs, and 4 of the genomes contain a little over 71 indicating some redundancy of these genomes. The unknown genome only contains 4 single copy core genes identified in this pipeline. This is a good check, as there are 71 single copy markers in the Bacteria_71 set, and they were all found within these sets of contigs.

I then wanted to compare the classification obtained through the Anvi'o workflow based on the taxonomy of the Ribosomal L2 gene and the given classification in the contigs file.

```
# import the taxonomy results
tax_table <- read.table("results/infant-gut-tax.txt", header=TRUE, sep="\t") %>%
  mutate(gene_callers_id = gsub("Ribosomal_L2_", "", scg_name)) %>%
  unite("taxonomy", 3:9, sep=";")

tax_table$gene_callers_id <- as.integer(tax_table$gene_callers_id)

# join with the gene calls table for the bin
left_join(tax_table, genes_table) %>%
  select(bin, taxonomy)
```

```
## Joining, by = "gene_callers_id"
```

```
##                                             bin
## 1       bin=0118A1023_Clostridium_difficile_28_5
## 2        bin=0118A1023_Citrobacter_koseri_53_430
## 3 bin=0118A1023_Bacteroides_vulgatus-like_42_520
## 4      bin=0118A1023_Enterococcus_faecalis_36_53
## 5                             bin=0118A1023_UNK
##
```

```
## 1 Bacteria;Firmicutes;Clostridia;Peptostreptococcales;Peptostreptococcaceae;Clostridioides;Clostridi
## 2                         Bacteria;Proteobacteria;Gammaproteobacteria;Enterobacterales;Enterobacteri
## 3                   Bacteria;Bacteroidota;Bacteroidia;Bacteroidales;Bacteroidaceae;Phocaeicola;Phoca
## 4                    Bacteria;Firmicutes;Bacilli;Lactobacillales;Enterococcaceae;Enterococcus;Entero
## 5 Bacteria;Desulfobacterota;Desulfovibrionia;Desulfovibrionales;Desulfovibrionaceae;Bilophila;Biloph
```

Comparing the results of the classification from the Ribosomal L2 hits with the preliminary classifications given in the contigs file, the classifications match very well accounting that there are taxonomical differences in species names such as "Bacteroides vulgatus" and "Phocaeicola vulgatus" because of differences between the NCBI and GTDB taxonomy systems. Additionally, based on the classification of the Ribosomal L2 gene within the `0118A1023_UNK` bin that did not contain a classification, it is classified as `Bilophila wadsworthia` within the Desulfobacterota, which in the NCBI taxonomy roughly equates to the Deltaprotoebacteria which is split into multiple phyla in the GTDB.

This quick sanity check ensures that 1) There are indeed approximately 5 bacterial "species" based on the single copy core gene analysis since only 5 Ribosomal L2 genes were identified and 2) The classification of those ribosomal L2 genes matches the classifications given for the bins in the contigs file. Additionally, this analysis provided a putative classification for the unknown bin. Furthermore, by running the `Bacteria_71` HMM collection on the contigs, we can tell that the 4 genomes besides the unknown/Desulfobacterota bin contain a high number of single copy core genes and are likely of higher quality. For now for downstream annotation and analysis we will keep the unknown/Desulfobacterota bin and take the results into account for the possible lower quality of this bin.

The correct and accurate analysis for this set of contigs would be to run GTDB-tk to confirm the classification and run checkM to assess completeness and contamination. However each of these tools requires a higher memory allocation and resources than quick checks with Anvi'o and for a quick spot check this analysis will suffice.

Read Coverage Inspection

Functional Annotation

Since the anvi'o pipeline already annotated ORFs with prodigal, we can export the proteins with `anvi-get-sequences-for-gene-calls -c contigs.db --get-aa-sequences -o infant-gut-proteins.faa` and then annotate using the KofamKOALA pipeline. This will annotate all proteins based on the KEGG database, from which I will then identify transposon families to investigate further, along with the gene neighborhoods/regions in which transposons are located.