METAGENOMICS PROJECT

Daniel Martín, Guillem Miró, Marc Pinós

2023-11-23

INTRODUCTION

Sequencing and metagenomics are biological disciplines that study the genetic and epigenetic sequence information of organisms. By sequencing their genome they attempt to understand several genetic concepts such as, genetic functionality, genotype, phenotype, and inheritance among others.

In this project a metagenomic analysis will be performed using an initial sequenced data set of an unknown bacterial species as reference. By performing an exhaustive methodology, latter sequence will be sorted out in order to identify its origin and produce several analysis.

Objectives

- To perform a metagenomic analysis from a genomic sequence of unknown origin.
- To select relevant and higher quality information from the sequence in order to properly work with it.
- To identify the taxonomic classification of the unknown sequence.
- To infer functional categories of the genomic protein sequences and analyze a selected category.
- To produce a phylogeny of the reference genome and its nearer species.

METHODS

Practical 4: Assembly of a bacterial genome

Raw sequence data will be assembled and analyzed so as to purge for a higher quality sample. In our particular case, sequencing data is composed in a FASTQ format, which provides the sequence itself and its quality information.

1. Read Quality Control

The sequence data will be analyzed by *Assess Read Quality with FastQC* so as to make en exhaustive report of the reads' quality. This will enable the detection of possible noise, unwanted/impure fragments, sequencing mistakes, and base distribution rates.

Parameters:

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores

- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence length distribution
- Sequence duplication levels

2. Trimming

Once all impure fragments are detected in the reads, trimming will be performed. By using *Trim Reads with Trimmomatic* an input read will be cut following a base pair (bp) threshold. For instance, starting and end point can be stipulated by selecting crop reads, which selects the number of bp to keep from start of the read.

The program will produce a new file which will correspond to the new sequence already trimmed.

3. Read Quality Control

The trimmed sequence will be assessed a new quality control (using *Assess Read Quality with FastQC* again) so as to identify all the compensations of quality compared to the raw analysis.

If all the previously-identified mistakes are compensated by the trimming, the analysis will be continued.

4. Assemble Read

All the reads will be stored into continuous sequences. In our case, *MEGAHIT* will be used. Compared to other softwares it is believed to be faster. All the reads will be conjugated to create long DNA fragments, known as *contigs* (assemble continuous genome fragments).

5. Binning

Previous *contigs* will be grouped in several *bins*. The term bin corresponds to a putative population genomes created by a selection of properties, such as codon usage, genome related statistics, GC proportion, read length, among other parameters. The analysis will be performed using *MaxBin2*.

6. Bin quality control

Finally, all bins created will be processed under a quality control. It will provide an estimate of genome completeness and contamination by plotting each bin compared to its expected distribution of a typical genome. *CheckM* will be used to perform the control.

In particular, we will look for high quality genomes, which correspond to >90% completeness and <5% of contamination.

Practical 5: Bacterial genome identification

Once all quality reads are selected, the analysis must begin. In particular, a taxonomic identification of out unknown sample will be performed.

In the case of bacteria, as they do not reproduce sexually, species classification is induced by the computation of ANI (average nucleotide identity). Bacteria are known to exchange core genes between mainly same species, and being unusual (or at least more difficult) to pass to different ones.

Therefore, computing percentage of identity between a reference species and the sample will enable an identity classification by gANI percentage: a) 95% same species; b) 85% same genus; c) 70% same family.

1. Genome annotation

The higher quality bin from practical 4 will be selected to perform the analysis. *PROKKA* software will be used to annotate the sample genome (placed in the bin) in order to identify several pieces or regions prone to be compared.

2. Genome set

Selecting latter annotated genome, we will create a genome set, enabilg a further research with the sample.

3. Microbe classification

The genome set will be taxonomically classified using GTDB. As an output a taxonomical tree will promote the identification of the bacterial sepecies present in our sample.

4. GTDB: genome taxonomy data base

GTDB will be used to download the *Genebank assembly accession* of out bacteria and acquire a FASTA reference sequence.

5. ANI computation

Latter FASTA sequence (type strain genome) and the annotated genome (Meta genome assembled genome) from step 1 will be compared using *FastANI*, which will compute the ANI.

In particular, an accurate ANI will be estimated by the comparison of nucleotide of orthologous gene pairs shared between the 2 sequences given.

Practical 6: Genome annotation I

In practical 6 we will proceed with the genome annotation of both sample and refernce genome.

1. Reference genome annotation

Following last practical's procedure, the genome extracted from GTDB will be also annotated.

2. Data exploration and functional categories

By selecting the annotated sample, we will select the binocular option in order to explore all data gathered.

Thanks to *PROKKA annotation pipeline* we will be able to acknowledge provenance, linked samples, genome overview, taxonomy, publications, and assembly and annotation from our sample. The latter category will be selected to find the functional categories encoded in our genome, which corresponds to a set of proteins prone to be expressed by our bacterial specimen.

The functional categories will be compared to the reference genome estracted from GTDB.

Afterwards, we will download a GFF (*gene feature format*) and a FASTA files to work on both genomes. A specific functional category will be selected and we will compute:

- The number of tRNAs of our sample.

- The number of rRNAs of our sample.
- The number of genes encoding proteins without known function.

Finally, using *RStudio* we will create a barplot of the types of genes found in our MAG and our genome of reference (type strain).

Practical 7: Genome annotation II

1. Multiple sequence alignment

Both nucleotide and amino acid sequences from our MAG and the selected reference genome will be aligned with MAFFT (Multiple Alignment using Fast Fourier Transform). The resulting file containing the alignment of both sequences will be visualized with Jalview.

2. Pairwise sequence alignment

In the same way as in the previous step, the pairwise alignment socre for both nucleotide and amino acid sequences from our MAG and the reference genome will be assessed using SMS (Sequence Manipulation Suite) from *bioinformatics.org.* This parameter takes into account the number of matches (i.e. A-A) and missmatches (i.e. A-T) between both sequences (as well as the possible gaps present in one of them) and computes them in order to give a value referring to their alignment level.

In our case Point accepted mutation (PAM) will be used to score the sequence alignments. PAM measures the amount of evolutionary distance between two amino acid sequences by the construction of PAM matrices. It uses PAM units matrices to determine such distance; for instance, one PAM unit matrix (PAM1 matrix) represents substitution probabilities for sequences that have experienced a single point mutation per 100 amino acids.

It is important to mention that PAM requires proteins which have high similarity with their predecessors. Moreover, PAM250 is frequently used for sequence comparison.

Practical 8: Building a phylogenetic tree

This last practical included in the project promotes the construction of a phylogenetic tree. A protein will be selected from the MAG's functional categories. Using a text editor, we will add a header of the gene code and protein name for its identification. Afterwards, its analysis will begin.

1. Homology search

Protein Blast will be used to search for protein sequence homologies with other characterized peptides. Among those, only 20 will be selected (from different species and genus) and downloaded. Such sequences will be stored in the same text file storing our reference protein.

2. Alignment and trimming

As we did in *Practical 7* MAFFT will be used to align all our 21 sequences. Then *Jalview* will be used to open the alignment and colour its residues. Observing the alignment some gappy regions might be identified. If so, they must be eliminated by trimming using *Trimal* software. The trimmed output will be also observed in *Jalview.*

3. Tree inference and visualisation

IQ-TREE software will be selected to construct a phylogenetic tree of the sequence aligned and trimmed file. Then, iTol will be used to visualize and modify the outputted tree.

All bootstrap values will be selected to be displayed and our reference protein will be highlighted.

RESULTS AND DISCUSSION

Practical 4

Upon acquiring the raw data for analysis, we conducted a thorough quality control assessment of all base pair positions within the short reads. Remarkably, we observed consistently high-quality scores across the entire span of the reads, including both head and tail fragments as depicted in *Figure 1*. Consequently, a decision was made to forgo Trimming procedures, aiming to preserve information from the extremities of the short reads.

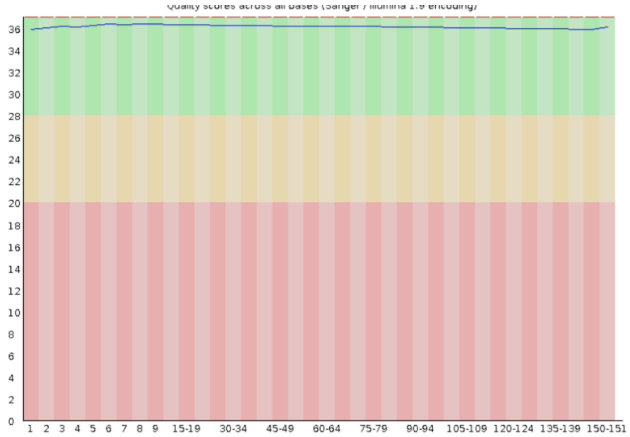


Figure 1.. Quality check reads scores

Following genome assembly, seven distinct bins were generated (*Figure 2*). Notably, bins 01, 03, and 04 exhibited completeness levels nearing maximum values, approximating 100%. This indicates the presence of all anticipated genes within the Metagenome-Assembled Genomes (MAGs).

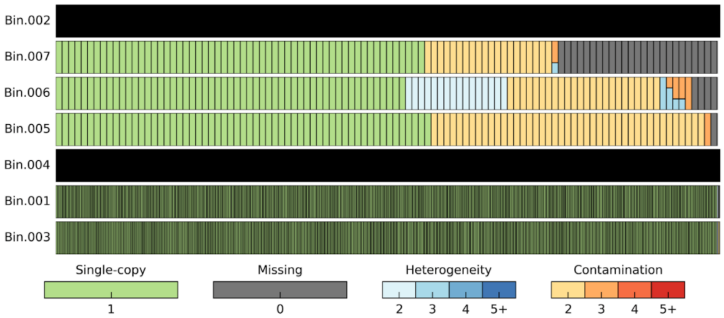
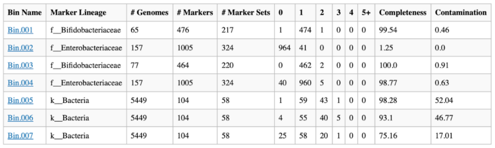


Figure 2. Bins generated after metagenome assembly

Additionally, these bins demonstrated a contamination level below 1%, affirming the appropriateness of the assembly process due to the initial high-quality short reads, enabling seamless assembly (*Figure 3*). Thus, bins 01, 03, and 04 can be considered of high quality, attaining >50% completeness and <5% contamination.

Upon analyzing the distribution of the short reads utilized for MAG assembly, it was observed that these ranged between 145-152 base pairs, slightly longer than the optimal size for assembly. Furthermore, approximately 5% of the sequences were identified as being repeated more than 10 times. Considering these findings, the GC content of our MAG might slightly deviate from the theoretical distribution (*Figure 4*).



Bin Name	Marker Lineage	# Genomes	# Markers	# Marker Sets	0	1	2	3	4	5+	Completeness	Contamination
Bin.001	f__Bifidobacteriaceae	65	476	217	1	474	1	0	0	0	99.54	0.46
Bin.002	f__Enterobacteriaceae	157	1005	324	964	41	0	0	0	0	1.25	0.0
Bin.003	f__Bifidobacteriaceae	77	464	220	0	462	2	0	0	0	100.0	0.91
Bin.004	f__Enterobacteriaceae	157	1005	324	40	960	5	0	0	0	98.77	0.63
Bin.005	k__Bacteria	5449	104	58	1	59	43	1	0	0	98.28	52.04
Bin.006	k__Bacteria	5449	104	58	4	55	40	5	0	0	93.1	46.77
Bin.007	k__Bacteria	5449	104	58	25	58	20	1	0	0	75.16	17.01

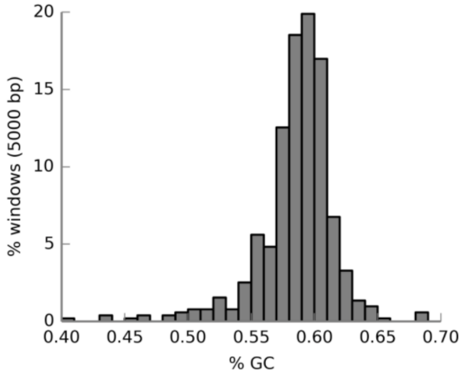
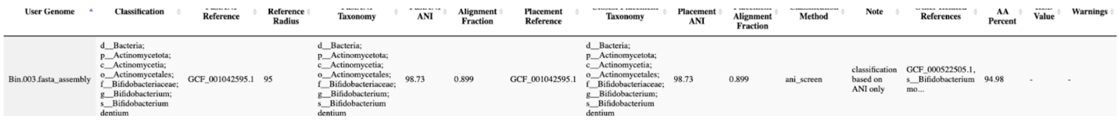


Figure 3 and 4. Bins completeness and contamination; GC content in bin 3

Practical 5

We opted to focus on bin03 due to its exceptional completeness level (100%) and minimal contamination (0.91%). Following the annotation of our MAG and subsequent genome set generation, we employed GTDB classification to determine the bacterial strain origin of this genome. The outcomes are presented in *Figure 5*, indicating a match between bin03 and *Bifidobacterium dentium* with an Average Nucleotide Identity (ANI) score of 98.73, as illustrated in the phylogenetic tree displayed in *Figure 6*. With a gANI value higher than 95%, we can infer that our MAG originates from *Bifidobacterium dentium*.



User Genome	Classification	Reference	Reference Radius	Taxonomy	ANI	Alignment Fraction	Placement Reference	Taxonomy	Placement ANI	Alignment Fraction	Method	Note	References	AA Percent	Value	Warnings
Bin.003.fasta_assembly	d__Bacteria; p__Actinomycetota; c__Actinomycetia; o__Actinomycetales; f__Bifidobacteriaceae; g__Bifidobacterium; s__Bifidobacterium dentium	GCF_001042595.1	95	d__Bacteria; p__Actinomycetota; c__Actinomycetia; o__Actinomycetales; f__Bifidobacteriaceae; g__Bifidobacterium; s__Bifidobacterium dentium	98.73	0.899	GCF_001042595.1	d__Bacteria; p__Actinomycetota; c__Actinomycetia; o__Actinomycetales; f__Bifidobacteriaceae; g__Bifidobacterium; s__Bifidobacterium dentium	98.73	0.899	ani_screen	classification based on ANI only	GCF_000522505.1, s__Bifidobacterium mo...	94.98	-	-

Figure 5. Bacterial Taxonomy classification

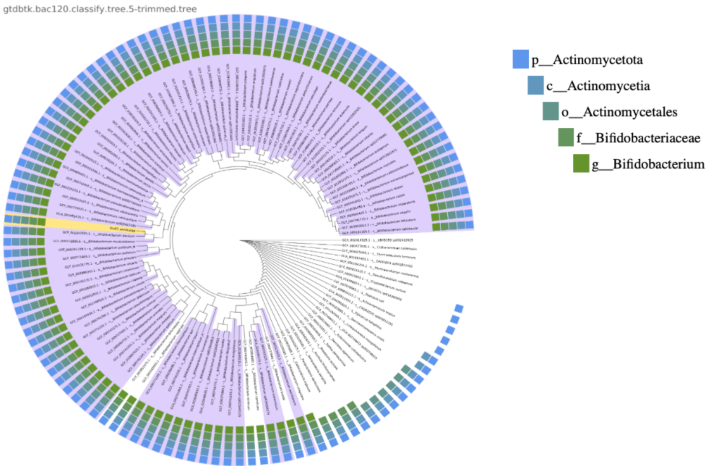
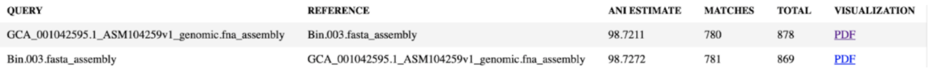


Figure 6. Phylogenetic tree

Subsequently, after obtaining the reference genome for *B. dentium*, we conducted a comparative analysis with our MAG, computing the nucleotide identity between both sequences. This assessment yielded an ANI distance estimate of 98.72% (*Figure 7*), indicating a similarity of 98.72% in nucleotide composition between the sequences. The comparison details can be visualized in *Figure 8*.



QUERY	REFERENCE	ANI ESTIMATE	MATCHES	TOTAL	VISUALIZATION
GCA_001042595.1_ASM104259v1_genomic.fna_assembly	Bin.003.fasta_assembly	98.7211	780	878	PDF
Bin.003.fasta_assembly	GCA_001042595.1_ASM104259v1_genomic.fna_assembly	98.7272	781	869	PDF

Figure 7. ANI distance estimation

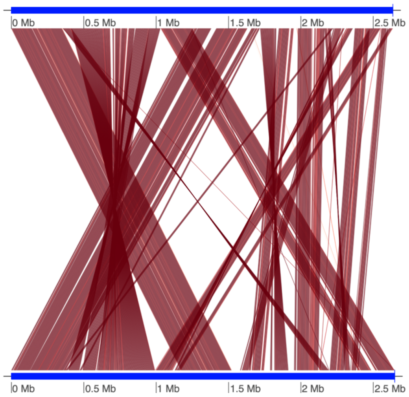
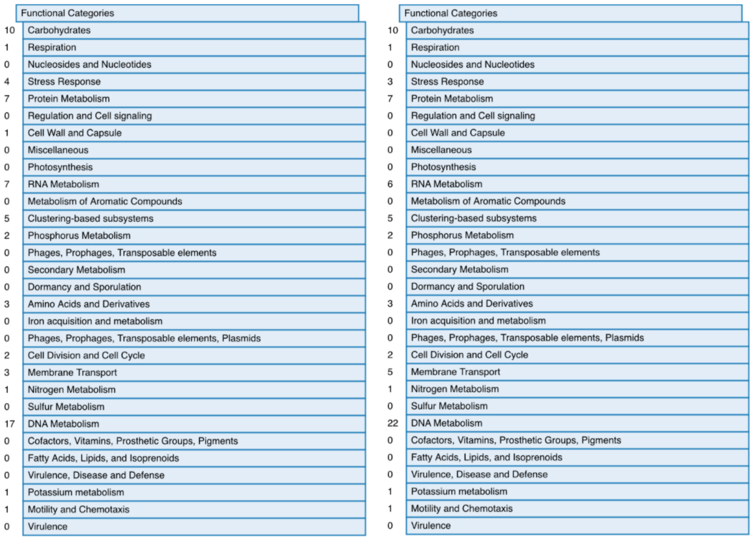


Figure 8. fastANI representation

Practical 6

Upon comparing the feature counts between our genome (Bin03) and the reference genome, we observed 2216 features in our genome as opposed to 2137 in the reference genome. Among these features, the functional categories encoded in both genomes are depicted in *Figure 9*. Notably, both genomes exhibit similar genes in most categories, except for: a) Stress response, b) Cell wall and capsule, c) RNA metabolism, d) Membrane transport, and e) DNA metabolism.



	Functional Categories
10	Carbohydrates
1	Respiration
0	Nucleosides and Nucleotides
4	Stress Response
7	Protein Metabolism
0	Regulation and Cell signaling
1	Cell Wall and Capsule
0	Miscellaneous
0	Photosynthesis
7	RNA Metabolism
0	Metabolism of Aromatic Compounds
5	Clustering-based subsystems
2	Phosphorus Metabolism
0	Phages, Prophages, Transposable elements
0	Secondary Metabolism
0	Dormancy and Sporulation
3	Amino Acids and Derivatives
0	Iron acquisition and metabolism
0	Phages, Prophages, Transposable elements, Plasmids
2	Cell Division and Cell Cycle
3	Membrane Transport
1	Nitrogen Metabolism
0	Sulfur Metabolism
17	DNA Metabolism
0	Cofactors, Vitamins, Prosthetic Groups, Pigments
0	Fatty Acids, Lipids, and Isoprenoids
0	Virulence, Disease and Defense
1	Potassium metabolism
1	Motility and Chemotaxis
0	Virulence

	Functional Categories
10	Carbohydrates
1	Respiration
0	Nucleosides and Nucleotides
3	Stress Response
7	Protein Metabolism
0	Regulation and Cell signaling
0	Cell Wall and Capsule
0	Miscellaneous
0	Photosynthesis
6	RNA Metabolism
0	Metabolism of Aromatic Compounds
5	Clustering-based subsystems
2	Phosphorus Metabolism
0	Phages, Prophages, Transposable elements
0	Secondary Metabolism
0	Dormancy and Sporulation
3	Amino Acids and Derivatives
0	Iron acquisition and metabolism
0	Phages, Prophages, Transposable elements, Plasmids
2	Cell Division and Cell Cycle
5	Membrane Transport
1	Nitrogen Metabolism
0	Sulfur Metabolism
22	DNA Metabolism
0	Cofactors, Vitamins, Prosthetic Groups, Pigments
0	Fatty Acids, Lipids, and Isoprenoids
0	Virulence, Disease and Defense
1	Potassium metabolism
1	Motility and Chemotaxis
0	Virulence

Figure 9. Functional categories encoded in MAG (left) and reference (right) genomes

Additionally, it is pertinent to ascertain the counts of tRNAs, rRNAs, and hypothetical coding genes present in our MAG compared to the reference. In our sample, the count stands at 46 for rRNAs, 6 for tRNAs, and 1059 for putative coding genes. These counts are contrasted with those of the reference in *Figure 10*.

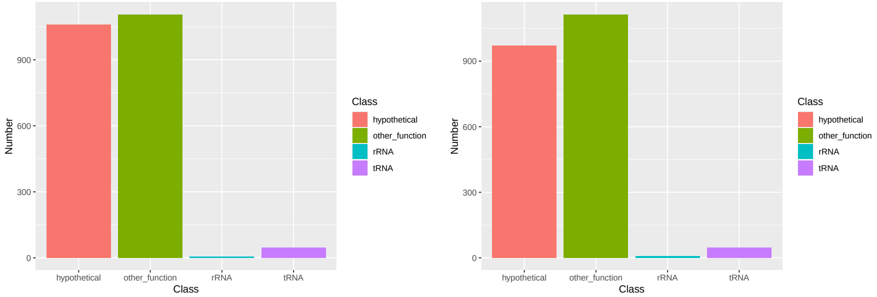


Figure 10. MAG (left) and reference (right) genes graphic

Practical 7

The gene selected is categorized within the "Clustering-based subsystems" gene group, known for its involvement in Bacterial Cell Division. This gene group encompasses a total of 5 coding genes in both our MAG and the reference genome. More precisely, the gene IGLBDLCO_01642 found in the MAG and PKGELIAN_01503 identified in the reference genome share a sequence length of 963 base pairs (bp). This sequence encodes a protein comprising 320 aminoacids,known as the Cell division protein FtsQ.

The multiple nucleotide sequence alignment between the MAG and the reference genome is depicted in *Supplementary Info. 1*, revealing 9 mismatches across the 963 bp sequence length. These mismatches are visually highlighted in *Figure 11*, displaying a gap in the consensus sequence (black box).

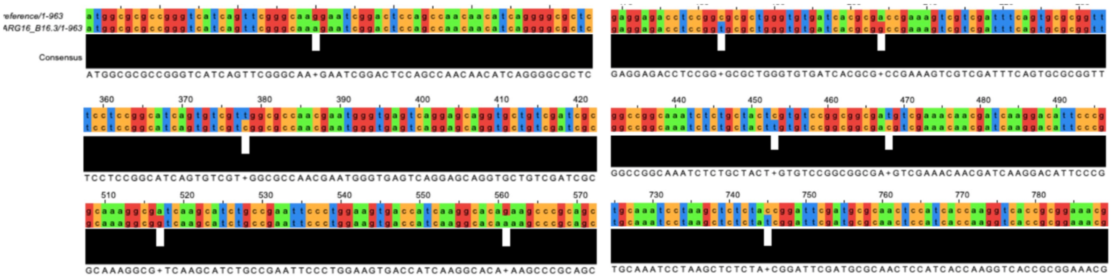


Figure 11. Missmatches between FstQ reference (up) and MAG (down) nucleotide sequence.

Notably, the multiple aminoacid alignment sequences is showcased in *Supplementary info 2* and more explicitly in *Figure 12*. Among the 9 previous mismatches, it's evident that only 2 involve a non-synonymous mutation in our MAG when compared to the reference:

- I73V: Considering the non-polar nature of both aminoacids, the substitution of isoleucine with valine is unlikely to significantly impact the functionality of the protein.
- P249S: Proline is non-polar, whereas serine is polar. This substitution may result in a change in functionality.

Finally, we conducted a pairwise nucleotide sequence alignment (not depicted) using the following values: match = 2, mismatch = -1 and gap = -2. This analysis yielded a score of 1899. Considering the sequence length of 963 nucleotides (nt), the maximum attainable score would be 1926 if all nucleotides were identical.

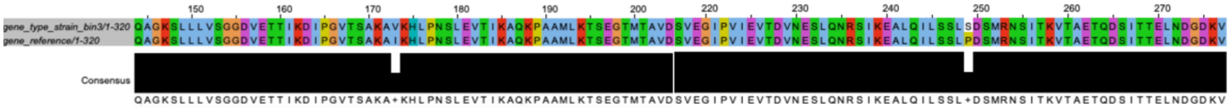

150
160
170
180
190
200
220
230
240
250
260
270
gene_type_strain_bin3/1-320
gene_reference/1-320
QAGKSLLLVSGGDVETTIKDIPGVTSAKAVKHLPNSLEVTIKAQKPAAMLKTSEGTMTAVDSVEGIPVIEVTDVNESLQNRSIKEALQILSSLSDSMRNSITKVTAETQDSITTELNDGDKV
QAGKSLLLVSGGDVETTIKDIPGVTSAKAIKHLPNSLEVTIKAQKPAAMLKTSEGTMTAVDSVEGIPVIEVTDVNESLQNRSIKEALQILSSLPDSMRNSITKVTAETQDSITTELNDGDKV
Consensus
QAGKSLLLVSGGDVETTIKDIPGVTSAKA+KHLPNSLEVTIKAQKPAAMLKTSEGTMTAVDSVEGIPVIEVTDVNESLQNRSIKEALQILSSL+DSMRNSITKVTAETQDSITTELNDGDKV

Figure 12. Non-synonymous mutatuions between FstQ MAG (up) and reference (down) aminoacid sequence.

Additionally, we performed a pairwise aminoacid sequence alignment using the PAM method (suitable for closely related protein sequences), resulting in a score of 1993.

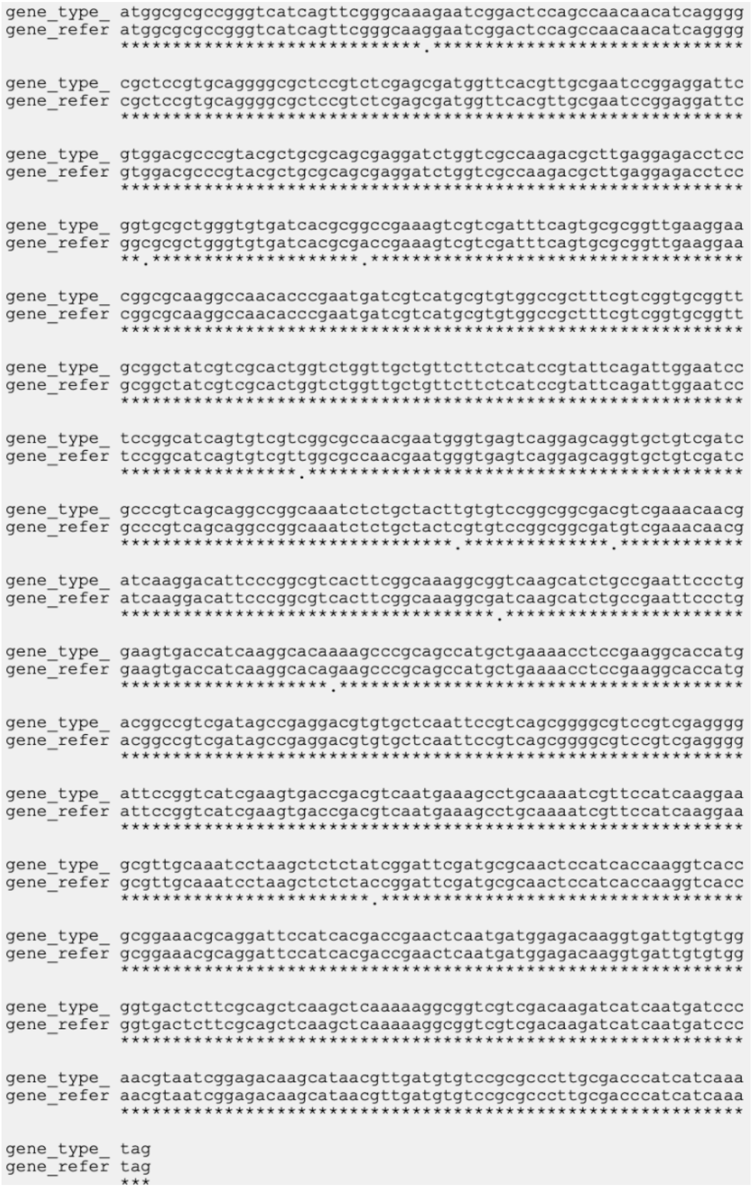
Based on these findings, it can be inferred that our MAG sequence encoding the FtsQ protein closely resembles the reference sequence. This suggests that our sample likely originates from *Bifidobacterium dentium* or a microorganism sharing a similar FtsQ protein sequence.

Practical 8

CONCLUSIONS

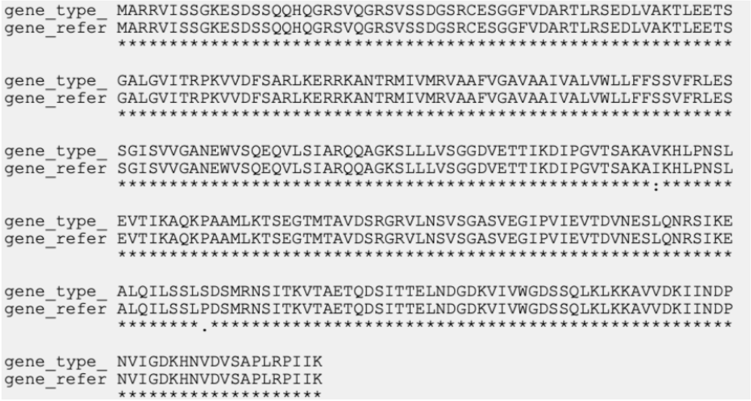
REFERENCES

SUPPLEMENTARY INFO



```
gene_type_ atggcgcgccgggtcatcagttcgggcaaagaatcggactccagccaacaacatcagggg
gene_refer atggcgcgccgggtcatcagttcgggcaaggaatcggactccagccaacaacatcagggg
           *****************************.******************************

gene_type_ cgctccgtgcaggggcgctccgtctcgagcgatggttcacgttgcgaatccggaggattc
gene_refer cgctccgtgcaggggcgctccgtctcgagcgatggttcacgttgcgaatccggaggattc
           ************************************************************

gene_type_ gtggacgcccgtacgctgcgcagcgaggatctggtcgccaagacgcttgaggagacctcc
gene_refer gtggacgcccgtacgctgcgcagcgaggatctggtcgccaagacgcttgaggagacctcc
           ************************************************************

gene_type_ ggtgcgctgggtgtgatcacgcggccgaaagtcgtcgatttcagtgcgcggttgaaggaa
gene_refer ggcgcgctgggtgtgatcacgcgaccgaaagtcgtcgatttcagtgcgcggttgaaggaa
           **.********************.************************************

gene_type_ cggcgcaaggccaacacccgaatgatcgtcatgcgtgtggccgctttcgtcggtgcggtt
gene_refer cggcgcaaggccaacacccgaatgatcgtcatgcgtgtggccgctttcgtcggtgcggtt
           ************************************************************

gene_type_ gcggctatcgtcgcactggtctggttgctgttcttctcatccgtattcagattggaatcc
gene_refer gcggctatcgtcgcactggtctggttgctgttcttctcatccgtattcagattggaatcc
           ************************************************************

gene_type_ tccggcatcagtgtcgtcggcgccaacgaatgggtgagtcaggagcaggtgctgtcgatc
gene_refer tccggcatcagtgtcgttggcgccaacgaatgggtgagtcaggagcaggtgctgtcgatc
           *****************.******************************************

gene_type_ gcccgtcagcaggccggcaaatctctgctacttgtgtccggcggcgacgtcgaaacaacg
gene_refer gcccgtcagcaggccggcaaatctctgctactcgtgtccggcggcgatgtcgaaacaacg
           ********************************.**************.************

gene_type_ atcaaggacattcccggcgtcacttcggcaaaggcggtcaagcatctgccgaattccctg
gene_refer atcaaggacattcccggcgtcacttcggcaaaggcgatcaagcatctgccgaattccctg
           ************************************.***********************

gene_type_ gaagtgaccatcaaggcacaaaagcccgcagccatgctgaaaacctccgaaggcaccatg
gene_refer gaagtgaccatcaaggcacagaagcccgcagccatgctgaaaacctccgaaggcaccatg
           ********************.***************************************

gene_type_ acggccgtcgatagccgaggacgtgtgctcaattccgtcagcggggcgtccgtcgagggg
gene_refer acggccgtcgatagccgaggacgtgtgctcaattccgtcagcggggcgtccgtcgagggg
           ************************************************************

gene_type_ attccggtcatcgaagtgaccgacgtcaatgaaagcctgcaaaatcgttccatcaaggaa
gene_refer attccggtcatcgaagtgaccgacgtcaatgaaagcctgcaaaatcgttccatcaaggaa
           ************************************************************

gene_type_ gcgttgcaaatcctaagctctctatcggattcgatgcgcaactccatcaccaaggtcacc
gene_refer gcgttgcaaatcctaagctctctaccggattcgatgcgcaactccatcaccaaggtcacc
           ************************.***********************************

gene_type_ gcggaaacgcaggattccatcacgaccgaactcaatgatggagacaaggtgattgtgtgg
gene_refer gcggaaacgcaggattccatcacgaccgaactcaatgatggagacaaggtgattgtgtgg
           ************************************************************

gene_type_ ggtgactcttcgcagctcaagctcaaaaaggcggtcgtcgacaagatcatcaatgatccc
gene_refer ggtgactcttcgcagctcaagctcaaaaaggcggtcgtcgacaagatcatcaatgatccc
           ************************************************************

gene_type_ aacgtaatcggagacaagcataacgttgatgtgtccgcgcccttgcgacccatcatcaaa
gene_refer aacgtaatcggagacaagcataacgttgatgtgtccgcgcccttgcgacccatcatcaaa
           ************************************************************

gene_type_ tag
gene_refer tag
           ***
```

Supplementary Info. 1 FstQ nucleotide sequence alignment between MAG (up) and reference (down).



```
gene_type_ MARRVISSGKESDSSQQHQGRSVQGRSVSSDGSRCESGGFVDARTLRSEDLVAKTLEETS
gene_refer MARRVISSGKESDSSQQHQGRSVQGRSVSSDGSRCESGGFVDARTLRSEDLVAKTLEETS
           ************************************************************

gene_type_ GALGVITRPKVVDFSARLKERRKANTRMIVMRVAAFVGAVAAIVALVWLLFFSSVFRLES
gene_refer GALGVITRPKVVDFSARLKERRKANTRMIVMRVAAFVGAVAAIVALVWLLFFSSVFRLES
           ************************************************************

gene_type_ SGISVVGANEWVSQEQVLSIARQQAGKSLLLVSGGDVETTIKDIPGVTSAKAVKHLPNSL
gene_refer SGISVVGANEWVSQEQVLSIARQQAGKSLLLVSGGDVETTIKDIPGVTSAKAIKHLPNSL
           ****************************************************:*******

gene_type_ EVTIKAQKPAAMLKTSEGTMTAVDSRGRVLNSVSGASVEGIPVIEVTDVNESLQNRSIKE
gene_refer EVTIKAQKPAAMLKTSEGTMTAVDSRGRVLNSVSGASVEGIPVIEVTDVNESLQNRSIKE
           ************************************************************

gene_type_ ALQILSSLSDSMRNSITKVTAETQDSITTELNDGDKVIVWGDSSQLKLKKAVVDKIINDP
gene_refer ALQILSSLPDSMRNSITKVTAETQDSITTELNDGDKVIVWGDSSQLKLKKAVVDKIINDP
           ********.***************************************************

gene_type_ NVIGDKHNVDVSAPLRPIIK
gene_refer NVIGDKHNVDVSAPLRPIIK
           ********************
```

Supplementary Info. 2 FstQ nucleotide sequence alignment between MAG (up) and reference (down).