# Gonococci assembly report

Daria Nemirich

March 2020

## Introduction

The report sums up the results of 16 *Neisseria gonorrhoeae* samples analysis. This includes assembly, annotation, single-nucleotide polymorphisms calling, antimicrobial resistance determinants and virulence factors search for each isolate.

Table 1: Sample metadata

| Sample | Strain | Phenotype |
|---|---|---|
| 1 | 07/15/03 | wt |
| 2 | 07/15/04 | wt |
| 3 | 07/15/59 | mut |
| 4 | 07/15/60 | mut |
| 5 | 07/16/41 | mut |
| 6 | 12/15/01 | wt |
| 7 | 12/15/02 | wt |
| 8 | 12/15/04 | mut |
| 9 | 12/16/14 | mut |
| 10 | 14/16/41 | wt |
| 11 | 19/16/03 | wt |
| 12 | 20/16/05 | wt |
| 13 | 20/16/25 | mut |
| 14 | 28/15/01 | wt |
| 15 | 41/16/11 | mut |
| 16 | 07/05/21 | wt |

## Reads quality check and trimming

Raw paired-end reads were acquired for each isolate. Prior to genome assembly, reads were trimmed with bbduk script from BBTools suite[1] with the following parameters:

- minimal average quality (maq) - 20
- k-mer trimming (ktrim) - r (from the right side)

- k-mer length (k) - 21
- minimal k-mer size (mink) - 11
- allowed number of mismatches (hdist) - 2
- quality trim threshold (trimq) - 20
- quality trim to Q20 from both ends (qtrim) - rl
- force-trimming of 20 leftmost (ftl) - 20
- read trimming to the same length (tpe)
- adapter trimming based on pair overlap detection (tbo)

Before and after the trimming procedure reads quality was evaluated with FastQC[2] software. Produced quality reports were aggregated across all samples for ease of use with MultiQC tool[3].

Table 2: *N.gonorrhea* reads trimming statistics

| Sample name | Input reads (in pairs) | Surviving (%) | Dropped (%) |
|---|---|---|---|
| 1 | 483,574 | 459,983 (95.2 %) | 23,591 (4.8 %) |
| 2 | 1,121,941 | 1,029,419 (91.5 %) | 95,522 (8.5 %) |
| 3 | 1,708,862 | 1,681,399 (98.4 %) | 27,463 (1.6 %) |
| 4 | 1,017,622 | 1,002,750 (98.5 %) | 14,872 (1.5 %) |
| 5 | 522,971 | 504,840 (96.5 %) | 18,131 (3.5 %) |
| 6 | 1,059,460 | 1,000,386 (94.4 %) | 59,074 (5.6 %) |
| 7 | 883,253 | 870,176 (98.5 %) | 13,077 (1.5 %) |
| 8 | 983,053 | 969,863 (98.7 %) | 13,190 (1.3 %) |
| 9 | 675,981 | 654,492 (96.8 %) | 21,489 (3.2 %) |
| 10 | 1,193,024 | 1,125,391 (94.3 %) | 67,633 (5.7 %) |
| 11 | 854,785 | 842,445 (98.6 %) | 12,340 (1.4 %) |
| 12 | 774,679 | 764,072 (98.6 %) | 10,607 (1.4 %) |
| 13 | 883,295 | 812,811 (92.1 %) | 70,484 (7.9 %) |
| 14 | 1,898,158 | 1,683,003 (88.7 %) | 215,155 (11.3 %) |
| 15 | 867,002 | 845,186 (97.5 %) | 21,816 (2.5 %) |
| 16 | 1,538,938 | 1,509,061 (98.1 %) | 29,877 (1.9 %) |

Trimmed reads can be found in the **trimmed** directory, FastQC and MultiQC reports for raw and trimmed reads are stored in **raw_qc** and **trimmed_qc** directories, respectively.

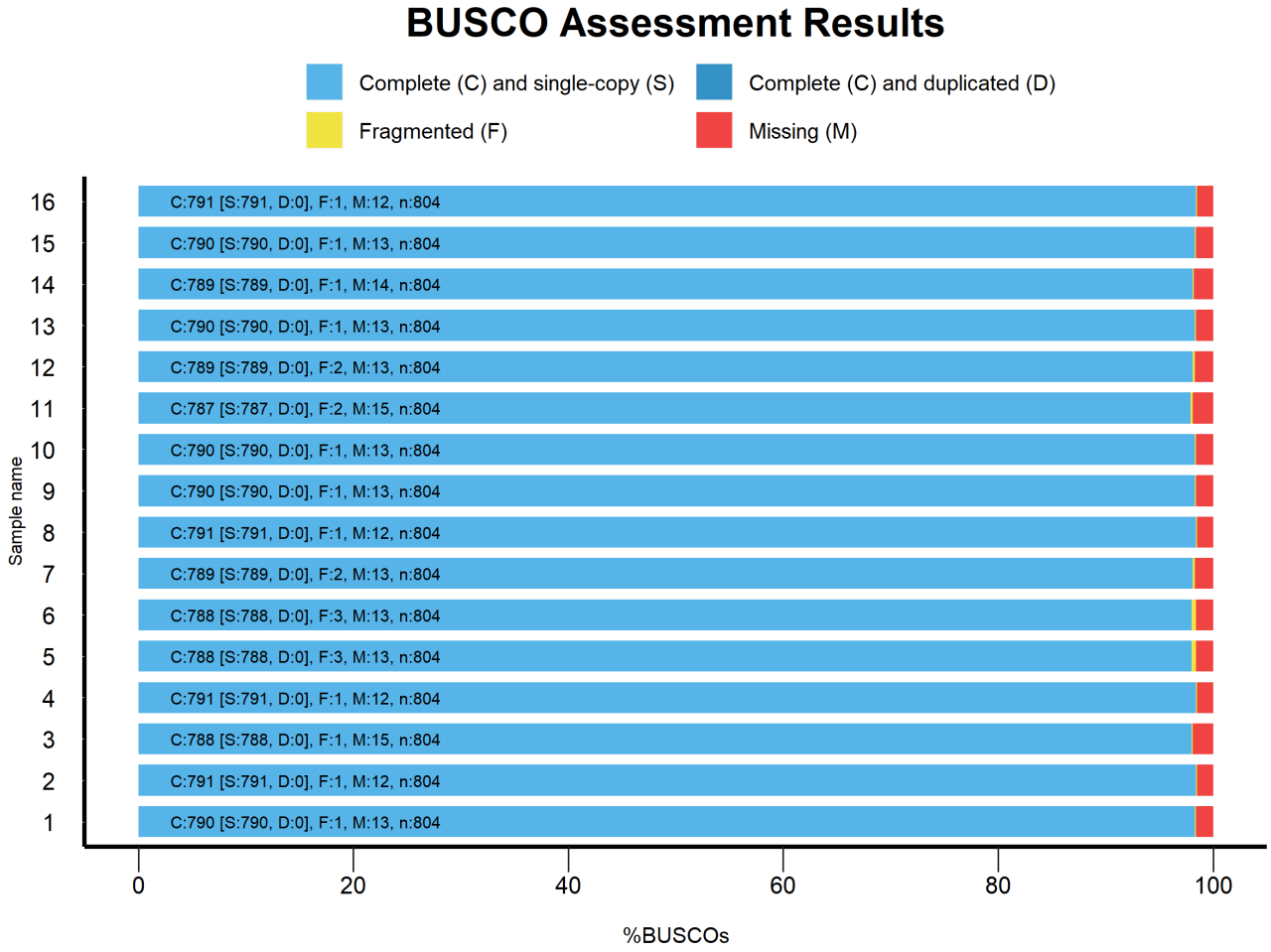
## Genome assembly and assembly quality control

Reads were assembled into contigs with SPAdes 3.13[4] with inner error correction process, performed by BayesHammer module, automatic selection of k-mers length and 'careful' option, which allows to minimize the number of mismatches in contigs. Final contigs for each isolate are stored separately in the corresponding folders in the **assembly** directory.

Table 3: Assemblies statistics

| Sample | Total assembly length | Number of contigs | Largest contig length | N50 | L50 | Number of misassemblies |
|---|---|---|---|---|---|---|
| 1 | 2,125,354 | 673 | 185,024 | 41,485 | 15 | 36 |
| 2 | 2,162,566 | 405 | 185,018 | 46,416 | 15 | 8 |
| 3 | 2,161,390 | 377 | 184,960 | 55,132 | 13 | 32 |
| 4 | 2,160,475 | 429 | 128,245 | 41,554 | 15 | 8 |
| 5 | 2,162,869 | 420 | 108,293 | 45,487 | 17 | 11 |
| 6 | 2,122,984 | 326 | 127,926 | 41,774 | 16 | 7 |
| 7 | 2,121,174 | 377 | 184,993 | 40,556 | 16 | 6 |
| 8 | 2,122,451 | 310 | 128,303 | 48,503 | 14 | 7 |
| 9 | 2,118,760 | 388 | 108,300 | 40,556 | 16 | 8 |
| 10 | 2,123,497 | 376 | 128,212 | 41,248 | 16 | 8 |
| 11 | 2,124,376 | 359 | 128,204 | 41,918 | 15 | 7 |
| 12 | 2,125,274 | 336 | 117,565 | 43,131 | 16 | 35 |
| 13 | 2,122,748 | 353 | 108,286 | 48,451 | 16 | 7 |
| 14 | 2,126,008 | 374 | 128,297 | 48,882 | 15 | 35 |
| 15 | 2,117,152 | 368 | 108,307 | 48,079 | 15 | 7 |
| 16 | 2,131,800 | 467 | 108,521 | 39,563 | 18 | 30 |

Assemblies quality was evaluated with QUAST 5.0.2[5] both with and without reference sequences. References sequences were selected with FastANI[6], an alignment-free whole-genome Average Nucleotide Identity (ANI) estimation algorithm, from all the assemblies available for *Neisseria gonorrhoeae* at NCBI Assembly database. Selected references are located in the **references_top** directory. Assemblies completeness was checked using BUSCO 4.0.5[7] software, which determines benchmarking unigene ortholog content.




Assemblies quality check statistics can be found in **assembly_qc**, **assembly_qc_reference_based** and **busco_qc** directories.

# Annotation

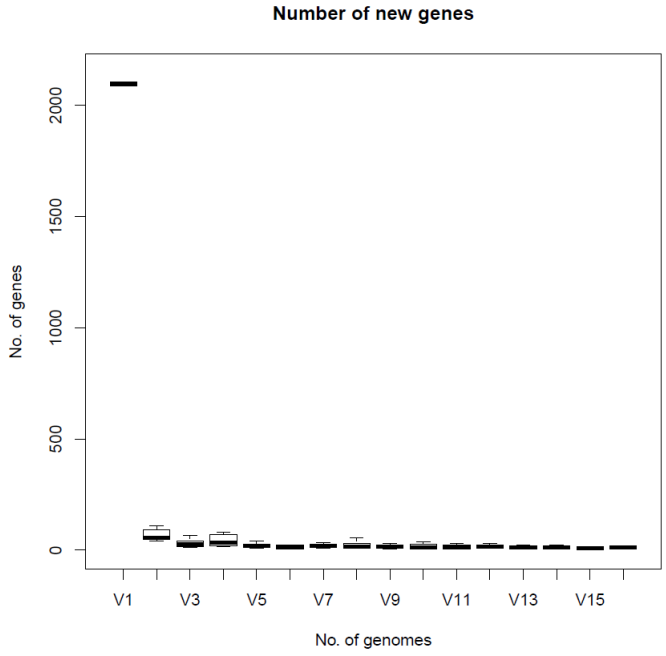
Assembled contigs were annotated using Prokka 1.12.[8]. Despite the availability of Prokka's *N.gonorrhoeae* database, a custom reference was created by gathering all *N.gonorrhoeae* protein entries from NCBI Identical Protein Groups (IPG) database in order to minimize the number of unidentified features. The file called **neisseria_ipg.fasta**, which contains IPG proteins, is available in the **internship** directory. Annotations for each isolate can found in the **annotation** directory placed in the respectively named subfolders. Notably, no CRISPR loci were identified by Prokka.

Table 4: Assemblies statistics

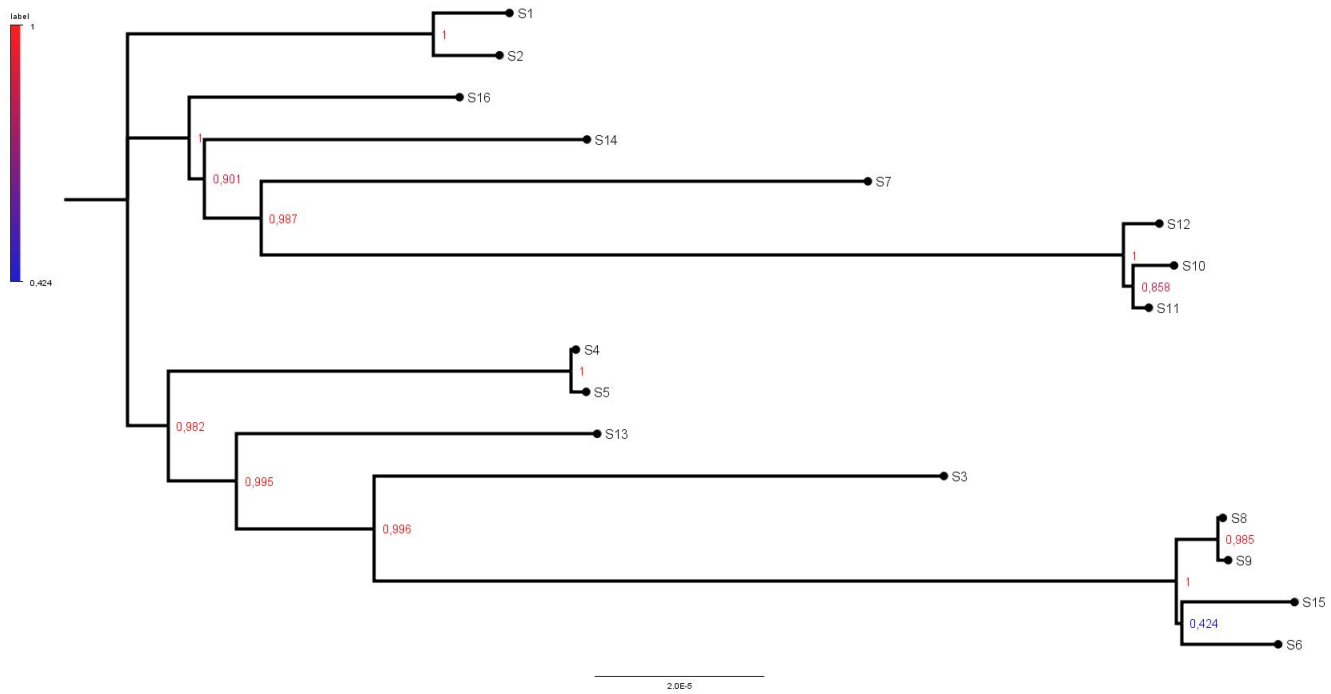
| Sample | CDS | rRNA | tRNA | tmRNA |
|---|---|---|---|---|
| 1 | 2150 | 3 | 55 | 1 |
| 2 | 2142 | 3 | 50 | 1 |
| 3 | 2141 | 3 | 51 | 1 |
| 4 | 2148 | 3 | 50 | 1 |
| 5 | 2146 | 3 | 51 | 1 |
| 6 | 2093 | 3 | 51 | 1 |
| 7 | 2092 | 3 | 51 | 1 |
| 8 | 2086 | 3 | 49 | 1 |
| 9 | 2094 | 3 | 49 | 1 |
| 10 | 2091 | 3 | 53 | 1 |
| 11 | 2091 | 3 | 50 | 1 |
| 12 | 2095 | 3 | 50 | 1 |
| 13 | 2098 | 3 | 50 | 1 |
| 14 | 2095 | 3 | 52 | 1 |
| 15 | 2091 | 3 | 50 | 1 |
| 16 | 2104 | 3 | 53 | 1 |

# Variant calling

In order to perform an SNP calling procedure, a reference genome from the available isolates was identified first. Annotated assemblies were used as an input for Roary 13.3.0[9], the pipeline that calculates pan-genome and produces core genes global alignment. Roary was launched with default parameters and two different alignment algorithms, namely PRANK and MAFFT.



The MAFFT-derived alignment was processed with Gblocks conserved blocks selector[10] for the sake of refinement. To infer approximately-maximum-likelihood phylogenetic tree from aforementioned blocks, FastTree[11] software was applied with Generalised time-reversible substitution model (GTR).




Phylogenetic tree visualization with FigTree 1.4.4[12]

Sample 1 was chosen as the reference for the variant calling procedure, based on the core genome phylogeny and Roary results. Thus, it was demonstrated that the first sample is characterized by a larger number of new genes (presumably due to a greater contig number) and is closer to the most recent common ancestor (MRCA) on the phylogenetic tree.

Trimmed reads were processed with Snippy 4.4.3 software[13] against the annotated S1 isolate assembly.Results of core genomes alignment, conserved blocks and phylogenetic tree can be found in **core_genomes_alngb** directory. Annotated SNP files are located in the **snp_calling** directory, each isolate in its own subfolder.

## Antibiotic resistance and virulence factors determinants search

In order to determine possible virulence factors, the latest version of Virulence Factors Database (VFDB)[14] was downloaded and only *N.gonorrhoeae* determinants were extracted. Then, blastp 2.2.31[15] search was performed with cut-off e-value at $10^{-5}$ and best hits were chosen based on the bit-score and e-value. After that hits were annotated back and blast tables were complemented with this information. All isolates possess virulence factors belonging to the following groups:

- Lipooligosaccharides: lgtA, lgtB, lgtC, lgtD, lgtE, lgtF, lgtG, rfaC, rfaF, rfaK;
- Type IV pili associated: pilC, pilD, pilE, pilF, pilG, pilH, pilI, pilJ, pilK, pilM, pilN, pilO, pilP, pilQ, pilT, pilT2, pilU, pilV, pilW, pilX, pilZ;
- Fatty acid resistance system: farA, farB;
- Multiple transferable resistance system: mtrC, mtrD, mtrE;
- Opacity-associated: opa, opc;
- Porins: porA, porB;
- Lactoferrin-binding protein: lbpA, lbpB.

Tab-separated tables for each isolate can be found in the **virulence_factors** directory. Each file represents blast search results table with an additional column containing virulence factors names and descriptions.

Antibiotics resistance determinants were estimated with Resistance Gene Identifier (RGI) software[16]. RGI was launched with the annotation-derived protein sequences and only strict hits permitted. Seven probable resistance genes were found for all 16 isolates.

Table 5: Identified antibiotic resistance determinants

| Gene | AMR gene family | Drug class | Resistance mechanism |
|---|---|---|---|
| mtrE | resistance-nodulation-cell division antibiotic efflux pump | macrolide antibiotic, penam | antibiotic efflux |
| mtrD | resistance-nodulation-cell division antibiotic efflux pump | macrolide antibiotic, penam | antibiotic efflux |
| mtrC | resistance-nodulation-cell division antibiotic efflux pump | macrolide antibiotic, penam | antibiotic efflux |
| macB | ATP-binding cassette antibiotic efflux pump | macrolide antibiotic | antibiotic efflux |
| macA | ATP-binding cassette antibiotic efflux pump | macrolide antibiotic | antibiotic efflux |
| farB | major facilitator superfamily antibiotic efflux pump | antibacterial free fatty acids | antibiotic efflux |
| farA | major facilitator superfamily antibiotic efflux pump | antibacterial free fatty acids | antibiotic efflux |

# Conclusion

16 *N.gonorrhoeae* isolates were analyzed concerning SNP, virulence factors determinants and genes providing antibiotics resistance. Despite the high similarity amongst all samples in terms of aforementioned features, sample number one has demonstrated the higher variability, presumably due to the greater contig number.

# References


[1] Brian Bushnel. *BBTools*. Joint Genome Institute. Berkeley, USA, 2017. URL: `https://sourceforge.net/projects/bbmap/`.

[2] Simon Andrews et al. *FastQC*. Babraham Institute. Babraham, UK, Jan. 2012. URL: `https://www.bioinformatics.babraham.ac.uk/projects/fastqc/`.

[3] P Ewels et al. "MultiQC: summarize analysis results for multiple tools and samples in a single report". In: *Bioinformatics* 32.19 (Oct. 2016), pp. 3047–3048. DOI: `10.1093/bioinformatics/btw354`.

[4] Sergey Nurk et al. "Assembling Genomes and Mini-metagenomes from Highly Chimeric Reads". In: *Research in Computational Molecular Biology*. Ed. by Minghua Deng et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 158–170. ISBN: 978-3-642-37195-0.

[5] Alexey Gurevich et al. "QUAST: quality assessment tool for genome assemblies". In: *Bioinformatics* 29.8 (Feb. 2013), pp. 1072–1075. DOI: `10.1093/bioinformatics/btt086`.

[6] Chirag Jain et al. "High-throughput ANI Analysis of 90K Prokaryotic Genomes Reveals Clear Species Boundaries". In: *Nature communications* (2018). DOI: `10.1038/s41467-018-07641-9`.

[7] Robert M Waterhouse et al. "BUSCO Applications from Quality Assessments to Gene Prediction and Phylogenomics". In: *Molecular Biology and Evolution* 35.3 (Dec. 2017), pp. 543–548. DOI: `10.1093/molbev/msx319`.

[8] Torsten Seemann. "Prokka: rapid prokaryotic genome annotation". In: *Bioinformatics* 30.14 (Mar. 2014), pp. 2068–2069. DOI: 10.1093/bioinformatics/btu153.

[9] Andrew J. Page et al. "Roary: rapid large-scale prokaryote pan genome analysis". In: *Bioinformatics* 31.22 (July 2015), pp. 3691–3693. DOI: 10.1093/bioinformatics/btv421.

[10] Gerard Talavera and Jose Castresana. "Improvement of Phylogenies after Removing Divergent and Ambiguously Aligned Blocks from Protein Sequence Alignments". In: *Systematic Biology* 56.4 (Aug. 2007), pp. 564–577. DOI: 10.1080/10635150701472164.

[11] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. "FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix". In: *Molecular Biology and Evolution* 26.7 (Apr. 2009), pp. 1641–1650. DOI: 10.1093/molbev/msp077.

[12] Andrew Rambaut. *FigTree*. 2009. URL: http://tree.bio.ed.ac.uk/software/figtree/.

[13] Torsten Seemann. *Snippy: fast bacterial variant calling from NGS reads*. 2015. URL: https://github.com/tseemann/snippy.

[14] Lihong Chen et al. "VFDB: a reference database for bacterial virulence factors". In: *Nucleic Acids Research* 33.suppl$_1$ (Jan. 2005), pp. D325–D328. DOI: 10.1093/nar/gki008.

[15] Stephen F. Altschul et al. "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs". In: *Nucleic Acids Research* 25.17 (Sept. 1997), pp. 3389–3402. DOI: 10.1093/nar/25.17.3389.

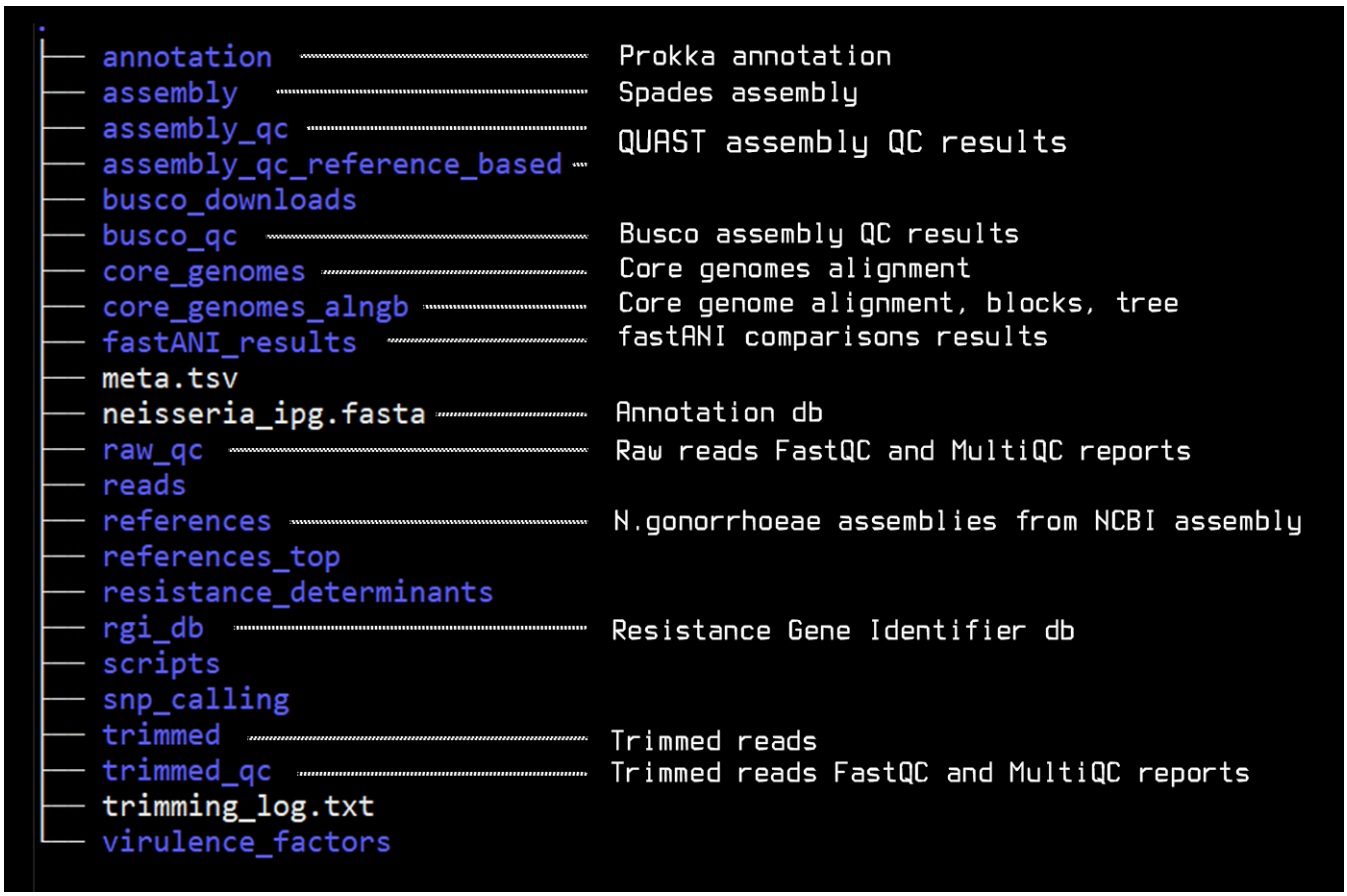
[16] Brian P Alcock et al. "CARD 2020: antibiotic resistome surveillance with the comprehensive antibiotic resistance database". In: *Nucleic Acids Research* 48.D1 (Oct. 2019), pp. D517–D525. DOI: 10.1093/nar/gkz935.

# Additional information

Three conda environments with the following packages were created for the analysis:

1. gonococci_assembly:
    - Python 3.8
    - FastQC 0.11.9
    - Spades 3.13.0
    - Prokka 1.12
    - Biopython
    - BBMap
    - MultiQC 1.8
    - Entrez-direct
2. assembly_qc
    - Python 3.6
    - QUAST 5.0.2
    - FastANI
    - Busco 4.0.5
    - Roary
    - Gblocks 0.91b
    - Snippy 4.4.3
3. rgi
    - RGI 4.2.2

Working directory has the following structure:




All necessary scripts are located in the **scripts** directory. There are:

- annotation.sh (Prokka annotation)
- assembly_qc.sh (Assembly QC performed with QUAST)
- assembly_qc_with_reference.sh (Assembly QC with the chosen reference performed with QUAST)
- busco_plots.sh (BUSCO plots with completed, fragmented and missed genes established)
- busco_qc.sh (Assembly QC performed with BUSCO)
- fastANI_search.sh (Comparison of the assemblies and references from the **references** directory to find the best reference for the QC procedure)
- get_references.sh (Assemblies download from the NCBI Assembly database)
- hits_annotation_launch.sh (This scripts launch the hits_annotation.py one)
- hits_annotation.py (Annotates blastp results of the virulence factors determination)
- reads_qc.sh (Performs raw reads QC with FastQC and MultiQC )
- reads_trimming.sh (Performs reads trimming and QC with FastQC and MultiQC)
- rgi_launch.sh (Launches resistance genes identifier)
- roary_launch.sh (Performs core genome assembly with Roary)
- snp_calling.sh (Executes variant calling with Snippy)
- vfs_search.sh (Performs blastp search of virulence factors in isolates and filtrates best hits)