



HAL
open science

Genomic pipeline for analysis of mutational events in bacteria

Pierre Lemée, Raphaël Charron, Arnaud Bridier

► **To cite this version:**

Pierre Lemée, Raphaël Charron, Arnaud Bridier. Genomic pipeline for analysis of mutational events in bacteria. Foodborne Bacterial Pathogens, In press. hal-04695580

HAL Id: hal-04695580

<https://hal.science/hal-04695580v1>

Submitted on 12 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Genomic pipeline for analysis of mutational events in bacteria

Pierre Lemée, Raphaël Charron, Arnaud Bridier

Running title: Analysis of adaptive mutations in bacteria

Abstract:

Unveiling the strategies of bacterial adaptation to stress constitute a challenging area of research. The understanding of mechanisms governing emergence of resistance to antimicrobials is of particular importance regarding the increasing threat of antibiotic resistance on public health worldwide. In the last decades, the fast democratization of sequencing technologies along with the development of dedicated bioinformatical tools to process data offered new opportunities to characterise genomic variations underlying bacterial adaptation. Thereby, research teams have now the possibility to dive deeper in the deciphering of bacterial adaptive mechanisms through the identification of specific genetic targets mediating survival to stress. In this chapter, we proposed a step-by-step bioinformatical pipeline enabling the identification of mutational events underlying biocidal stress adaptation associated to antimicrobial resistance development using *Escherichia marmotae* as an illustrative model.

Keywords: Bacterial adaptation, antimicrobial resistance, variant, whole genome sequencing, variant calling, mutation

1. Introduction

Microbial adaptive evolution have shaped, and continue to drive, the microbial communities and their impacts in pristine and anthropized environments worldwide. Bacterial ecosystems are mostly very competitive and bacteria undergoing a mutational event conferring them an advantage in specific conditions have the potential to outcompete their unmutated counterparts. Mutations improving nutrient absorption or metabolization [1] or bettering survival upon stress exposure, for example to an antibiotic and/or a biocide, have already been reported for instance [2–4]. Bacteria can evolve quickly, since some species are able to double their population size in less than 10 minutes [5]. Consequently, it is possible to study the genetic adaptation and evolutionary drift of the population upon various exposure conditions using laboratory-scaled experiences known as adaptive laboratory experiments (ALE) [6]. Some works thereby demonstrated that bacterial populations evolve continuously, even under stable environmental conditions, and that new mutational events recurrently occur leading to the propagation of mutants with increased fitness [7, 8]. Lenski demonstrated that even after 25 years of cultivation and 60000 generations, new variants with new genome modifications are still appearing and can exhibit an increased fitness and spread in the bacterial population [9].

ALE have been harnessed in various fields to decipher bacterial adaptive strategies. For instance, developments of dedicated experimental setups have been conducted to investigate the effects of

various drug concentrations on antimicrobial resistance emergence [10]. Such approaches enabled the identification of mechanisms involved in adaptation and resistance emergence, highly supported by the democratization of bacterial genome sequencing and analysis [2]. Since the discovery of the Sanger method in the 1970's [11], numerous innovative technologies have indeed been developed. Initially restrained to the analysis of specific genes in the genome, next-generation sequencing (NGS) enables today the reliable and accurate analysis of complete genomes of bacterial strains through an affordable and fast process [12]. Such technological advances highly contributed these last years to the identification of genetic targets governing bacterial adaptation in various environments and under different stresses (including antimicrobials). Whole genome sequencing (WGS) and comparative genomic approaches indeed provide the possibility to identify specific mutations between a wild-type (*parental strain*) and an adapted derivative strain (called hereafter *variant strain*) thus revealing the metabolic pathways involved in adaptation processes.

As an illustration, we presented in this chapter the bioinformatical pipeline of analysis developed in the frame of BAoBAb project (<https://anr.fr/Project-ANR-21-CE35-0001>). Using ALE combined with WGS, the objective of the project was to investigate the mechanism of bacterial adaptation to biocides by comparing genomes of parental strains and biocide-adapted variants to identify mutational events involved in biocidal stress adaptation.

2. Materials

Whole genome sequencing analysis requires several resources. First, to be able to run all the software, sufficient computing resources are needed. All these resources can be found in a public remote computing cluster. It is important to note that all the methodology presented is carried out in a UNIX environment using Bash consoles. Knowledge of this environment and of the Bash language is therefore necessary to carry out this analysis. Analyses must be performed on DNA from pure strain cultures with no contaminants. Sequencing is then achieved using Illumina whole genome sequencing instrument. For each strain, a pair of fastq files containing small reads (150nt) will be generated and used (see **Note 1**).

Various software packages are used in the analysis and must be installed before starting the analysis (**Table 1**). All these software should preferably be installed in different mamba (see **Note 2**) environments to avoid package versioning and conflicts.

3. Methods

The bioinformatical method described here enable to highlight mutations through the comparison of bacterial genomes. The genome of a parental strain not exposed to the stress (biocides, antibiotics, chemicals) is compared to the genome of the same strain having undergone a phase of exposure to stress which potentially led to adaptive mutations (*variant strain*). Effects of mutations found in genes or in a family of genes linked to a specific metabolic pathway can then be deeply investigated to understand biological mechanisms at the origin of the adaptation to stress in variant.

Bioinformatical analysis begins with the raw reads processing (trimming and filtering) for both parental and variant strains. Then a *de novo* assembly is carried out from the parental strain reads, as well as a gene annotation of the newly constituted genome to constitute a reference genome. Finally, processed raw reads of variants are aligned on the annotated reference genome of the

parental strain to identify discrepancies between parental and variant genome strains (mutational events) (**Figure 1**) (see **Note 3**).

3.1. Raw reads processing

Raw reads processing filters out reads of poor quality and eliminates any adapters (see **Note 4**) used during sequencing. The fastq file obtained after sequencing contains all the reads. These reads are associated with a quality score generated by the sequencer which will be used by the fastp software to eliminate bad quality reads. This step is done in the same way with the reads of the parental strain and the reads of the mutated strains.

```
fastp -i $FASTQ_R1 -I $FASTQ_R2 -o $FASTQ_CLEAN_R1 -O $FASTQ_CLEAN_R2 --unpaired1
$FASTQ_UNPAIRED --unpaired2 $FASTQ_UNPAIRED -h $FASTQ_REPORT_HTML --
detect_adapter_for_pe -M 25 -5 -r --correction
```

Where:

- `-i` is an option to input the first fastq reads file of a strain `$FASTQ_R1`.
- `-I` is an option to input the second fastq reads file of a strain `$FASTQ_R2`.
- `-o` is an option to output the first fastq reads file processed by fastp `$FASTQ_CLEAN_R1`.
- `-O` is an option to output the second fastq reads file processed by fastp `$FASTQ_CLEAN_R2`.
- `--unpaired1,2` are two options to output reads that are no longer paired with their second reads. For example, read1 passed quality control but not read2, read1 will not be paired anymore. All unpaired reads are stocked in the same file `$FASTQ_UNPAIRED`.
- `-h` is an option to output quality control report of the reads in HTML file `$FASTQ_REPORT_HTML`.
- `--detect_adapter_for_pe` is an option to detect and remove automatically sequencing adapters. It is only used for paired data.
- `-M` is an option to specify the quality threshold for `-5` and `-r` options. Here the threshold is setup on 25.
- `-5` is an option to drop the bases with bad quality at the start of the read. It uses a sliding window and cuts if the mean quality of the bases in the window are below the thresholds specified by the option `-M`.
- `-r` is almost the same option as `-5`. The sliding window cuts the right part of the read if the mean quality of the bases in the window are below the thresholds specified by the option `-M`.
- `--correction` is an option to enable base correction in overlapped regions.

The reads files are now properly filtered and cleaned. They can now be used either to assemble the reference genome with the reads of the parental strain, or to call variants with the mutated strains.

3.2. Parental strain genome

A *de novo* assembly can be made from the parental strains reads. This assembly will then be followed by a genome annotation from online databases. Finally, a quality control of this assembly will be established.

3.2.1. *De novo* assembly

De novo assembly is done using the Unicycler software.

```
unicycler -1 $PARENTAL_FASTQ_CLEAN_R1 -2 $PARENTAL_FASTQ_CLEAN_R2 -s $FASTQ_UNPAIRED -o $ASSEMBLY_FOLDER --min_fasta_length 200
```

Where:

- `-1` is an option to input the first fastq reads file of the parental strain `$PARENTAL_FASTQ_CLEAN_R1`.
- `-2` is an option to input the second fastq reads file of the parental strain `$PARENTAL_FASTQ_CLEAN_R2`.
- `-s` is an option to input the unpaired reads file of the parental strain `$FASTQ_UNPAIRED`.
- `-o` is an option to output all the result in a directory `$ASSEMBLY_FOLDER`. The `$PARENTAL_GENOME_FASTA` assembly in fasta format is stored in this directory under the name "assembly.fasta".
- `--min_fasta_length` is an option to specify the minimum length of a contig that makes up the assembly. Here, below 200nt, contigs are removed from the assembly.

The output folder of Unicycler contains the newly formed assembly composed of contigs of different sizes. Now the genome has to be annotated to get information about the genes composing it.

3.2.2. Genome annotation

Genome annotation is done with the Bakta software. It uses a database carrying information on the genes of different bacterial species. We will therefore first have to download the database before launching Bakta (see **Note 5**).

```
bakta_db download --output $BAKTA_DATABASE --type full
```

Where:

- `download` is an option to download the Bakta database.
- `--output` is an option to specify the output directory of the Bakta database `$BAKTA_DATABASE`.
- `--type` is an option to specify the database to download (full, light).

Now that the Bakta database has been downloaded, annotation can be carried out on the genome of the parental strain. It is possible to specify the bacterial species of the samples as an argument to facilitate annotation (see **Note 6**). Here, the species *Escherichia marmotae* will be indicated.

```
bakta --db $BAKTA_DATABASE --output $ANNOTATION_FOLDER --prefix PARENTAL --genus Escherichia --species marmotae $PARENTAL_GENOME_FASTA
```

Where:

- `--db` is an option to specify the directory of the Bakta database `$BAKTA_DATABASE`.
- `--output` is an option to specify the output directory of the Bakta annotation. `$ANNOTATION_FOLDER`. This folder will contain all the annotation files.
- `--prefix` is an option to specify the start of the name of each file in annotation directory.
- `--genus`, `--species` are options to specify the bacterial species of the parental strain. This option optimises the annotation according to the species provided.
- `$PARENTAL_GENOME_FASTA` is the input fasta file for the genome assembly contained in the `$ASSEMBLY_FOLDER` folder under the name "assembly.fasta".

The genome of the parental strain is annotated, and all that remains is to assess the quality of the assembly. The annotations obtained are also used in this evaluation to obtain as much information as possible.

3.2.3. Assembly quality

Quality of the assembly is assessed using Quast software. It will not directly indicate whether the assembly is of good or poor quality, but rather provides information about its content. For example, it can give information about the number of contigs and the size of the longest contig. The longer the contigs, the fewer they will be, and the better will be the quality of the assembly.

```
quast -o $ASSEMBLY_QUALITY_FOLDER -g $PARENTAL_ANNOTATION_GFF -1
$PARENTAL_FASTQ_CLEAN_R1 -2 $PARENTAL_FASTQ_CLEAN_R2 --single $FASTQ_UNPAIRED
$PARENTAL_GENOME_FASTA
```

Where:

- `-o` is an option to specify the output directory for Quast analyses.
- `-g` is an option to specify the annotation file `$PARENTAL_ANNOTATION_GFF` contained in the `$ANNOTATION_FOLDER` folder in .gff format.
- `-1`, `-2` are options to specify the two fastq reads files of the parental strain `$PARENTAL_FASTQ_CLEAN_R1` and `$PARENTAL_FASTQ_CLEAN_R2`.
- `--single` is an option to specify the fastq file of unpaired reads `$FASTQ_UNPAIRED`.
- `$PARENTAL_GENOME_FASTA` is the genome assembly contained of the parental strain.

A report is provided in the Quast results file. This contains all the statistics needed to establish the quality of the assembly made from the reads of the parent strain. The reference genome used to compare the mutated strains is now ready for the next step: variant calling from the genome of the parental strain.

3.3. Variant calling

Variant calling will be used to detect spontaneous mutations in stress-exposed strains. This is done by aligning the reads of mutated strains with the reference genome of the parental strain. The most identified variants are single nucleotide polymorphism (SNP). Insertions and deletions of a few nucleotides can also be detected. Snippy software is used for variant calling. The reads of the mutated strains that have been filtered in the fastp step can now be used. This process must be carried out for each stress-exposed strain to be analysed.

```
snippy --outdir $VARIANT_CALLING_FOLDER --prefix MUTATED_STRAIN --ref
$PARENTAL_ANNOTATION_GBFF --R1 $MUTATED_FASTQ_CLEAN_R1 --R2
$MUTATED_FASTQ_CLEAN_R2
```

Where:

- `--outdir` is an option to specify the output directory `$VARIANT_CALLING_FOLDER` containing all the variants.
- `--prefix` is an option to specify the start of the name of each file in the output directory. It is important to indicate the exact strain in this parameter to identify the files to be analysed for each strain.
- `--ref` is an option to specify the annotated genome file `$PARENTAL_ANNOTATION_GBFF` used as a reference in the analysis. This .gbff file is in the folder `$ANNOTATION_FOLDER`.

- `--R1`, `--R2` are options to specify the two fastq reads files of the mutated strain `$MUTATED_FASTQ_CLEAN_R1` and `$MUTATED_FASTQ_CLEAN_R2`.

The mutations found by snippy are stored in files in the output folder. It is now possible to analyse the variants for each mutated strain.

3.4. Analysis of the results – the example of BAoBAb project

Based on all the previous analyses, we can now study the results of the variants obtained for each mutated strain. Variants can be found either in genes directly or in intergenic areas. However, they can also be found in hypothetical proteins that were located during the initial stages of genome annotation but were subsequently not identified in the databases.

As an example, we will study a strain that was used in the BAoBAb project. Biofilms of an *Escherichia marmotae* strain (named Ec219) were exposed to various biocides for 4 weeks. Resistant variants to antibiotics emerging in biofilms were selected by plating on rifampicin-supplemented plates at various times upon exposure. Finally, the whole genome of the resistant variants was sequenced through short-reads technology and compared to the genome of the parental strain of *E. marmotae*. Mutations in these different variants can thereby be analysed, in the objective of characterizing mechanisms of cross-resistance between antibiotics and biocides.

Two variants from the study were selected here as illustrative examples (**Figure 2**):

- Ec219_W3_Rif100R_H2O: a resistant variant emerged after 3 weeks of exposure in Ec219 biofilms exposed to H₂O and selected on 100 mg/l rifampicin supplemented plates
- Ec219_W4_Rif20R_TMN: a resistant variant emerged after 4 weeks of exposure in Ec219 biofilms exposed to N-(3-aminopropyl)-N-dodecylpropane-1,3-diamine and selected on 20 mg/l rifampicin supplemented plates

The genome of the parental strain Ec219 was assembled and annotated using respectively Unicycler (3.2.1) and Bakta (3.2.2). After the analysis of the assembly quality using Quast (3.2.3), we ended up with a genome that had a final size of 4,531,508 bp. The genome was made up of 48 contigs with the largest contig being 632,041 bp in size and an N50 of 322,084 bp. The N50 is a measure of the size of the contig halfway through the genome. All contigs larger than or equal to the N50 constitute 50% of the genome. The average sequencing depth was 189. Depth is an important measure because it indicates the average number of reads stacked on top of each other for a given nucleotide. The greater the depth, the more precise is the information contained in the genome. If a mutation is found in a low depth zone (<10), then it is difficult to conclude on the truthfulness of the mutation as it may be due to a sequencing error.

Both variants exhibited each two mutations (**Table 2**).

Ec219_W3_Rif100R_H2O is mutated in the *rhsR* gene, encoding a transcriptional repressor for the ribose rbsDACBK operon, with a deletion of a guanine (G) at position 284,902 of the 7th contig of the genome. This mutation causes a shift in the reading frame of the protein. This type of mutation mainly causes the appearance of an early stop codon in the protein, resulting in partial or total loss of the protein's activity. The second mutation in the Ec219_W3_Rif100R_H2O variant is in the *rhoB* gene encoding the beta subunit of DNA-directed RNA polymerase and already described to be involved in rifampicin resistance. It is a single nucleotide polymorphism (snp), meaning that there has been a change from a cytosine to a thymine nucleotide at position 5,608 at the 22nd contig. This mutation causes a change in the code of the three-nucleotide codon, resulting in a change of an

amino acid in the final protein. Depending on the polarity of this amino acid compared with the base amino acid, the protein's conformation and affinity in the active site may be affected.

Ec219_W4_Rif20R_TMN shows a snp in the *rpoD* gene encoding the RNA polymerase sigma factor. There is a guanine to cytosine nucleotide change at position 428,187 of the second contig, also causing an amino acid change. The second mutation is located in an intergenic region between two genes. Nevertheless, it can still have an effect as it could be located on a promoter or suppressor site of a gene. In conclusion, different types of mutation can emerge, and it is important to analyse all of them carefully to draw out all the possible interpretations.

4. Notes

1. During the presentation of the method, the files and folders used during the analysis are all named in the form of the variable `$FILE_NAME`. Each call to this variable corresponds to the same file or folder.
2. Mamba is a conda extension that works in the same way as conda. However, it is an optimized version of conda and runs faster for each command. All the software used can be installed via conda and therefore via mamba.
3. It is important to differentiate sequential reads from parental and mutated strains. Despite the similarity of the analyses at the beginning of the pipeline, the reads will not be used by the same tools.
4. Adapters are short nucleotide sequences that are added to reads. They enable the sequences to bind with the flow cell (the support used for sequencing). They also enable sequences to be enriched via PCR. In some cases, they can be used to index different samples, enabling sequencing with mixed DNA.
5. Bakta has two databases that can be installed via a command: full or light. The full database is downloaded in the presented method to have the best annotation results. However, this database is very heavy in terms of storage. So, if you are having trouble installing the database it is better to download the light database.
6. In the Bakta options, it is possible to enter information about a bacterial species to facilitate annotation of the genome. It is also possible to add the `--strain` option to provide information at strain level.

5. Data availability

All the data used as examples in this work are available on NCBI in the BioProject [PRJNA947807](https://www.ncbi.nlm.nih.gov/bioproject/PRJNA947807).

Acknowledgements

This research was funded by l'Agence Nationale de la Recherche (ANR), project JCJC BAoBAb (ANR-21-CE35-0001). RC is a recipient of an Anses-INRAE doctoral fellowship.

French general directorate for food (DGAL) and the National Reference Laboratory "Résistance Antimicrobienne" hosted in the Anses laboratory of Fougères were thanked for the availability of *E. marmotae* strain.

References

1. D'Souza G, Kost C (2016) Experimental Evolution of Metabolic Dependency in Bacteria. *PLoS Genet* 12:e1006364. <https://doi.org/10.1371/journal.pgen.1006364>
2. Marciano DC, Wang C, Hsu T-K, Bourquard T, Atri B, Nehring RB, Abel NS, Bowling EA, Chen TJ, Lurie PD, Katsonis P, Rosenberg SM, Herman C, Lichtarge O (2022) Evolutionary action of mutations reveals antimicrobial resistance genes in *Escherichia coli*. *Nat Commun* 13:3189. <https://doi.org/10.1038/s41467-022-30889-1>
3. Douarre P-E, Sévellec Y, Le Grandois P, Soumet C, Bridier A, Roussel S (2022) FepR as a Central Genetic Target in the Adaptation to Quaternary Ammonium Compounds and Cross-Resistance to Ciprofloxacin in *Listeria monocytogenes*. *Front Microbiol* 13:864576. <https://doi.org/10.3389/fmicb.2022.864576>
4. Charron R, Lemée P, Huguet A, Minlong O, Boulanger M, Houée P, Soumet C, Briandet R, Bridier A (2023) Polyhexamethylene biguanide promotes adaptive cross-resistance to gentamicin in *Escherichia coli* biofilms. *Front Cell Infect Microbiol* 13:1324991. <https://doi.org/10.3389/fcimb.2023.1324991>
5. Weissman JL, Hou S, Fuhrman JA (2021) Estimating maximal microbial growth rates from cultures, metagenomes, and single cells via codon usage patterns. *Proc Natl Acad Sci U S A* 118:e2016810118. <https://doi.org/10.1073/pnas.2016810118>
6. Bennett AF, Hughes BS (2009) Microbial experimental evolution. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology* 297:R17–R25. <https://doi.org/10.1152/ajpregu.90562.2008>
7. Wisner MJ, Ribeck N, Lenski RE (2013) Long-term dynamics of adaptation in asexual populations. *Science* 342:1364–1367. <https://doi.org/10.1126/science.1243357>
8. Lenski RE, Wisner MJ, Ribeck N, Blount ZD, Nahum JR, Morris JJ, Zaman L, Turner CB, Wade BD, Maddamsetti R, Burmeister AR, Baird EJ, Bundy J, Grant NA, Card KJ, Rowles M, Weatherspoon K, Papoulis SE, Sullivan R, Clark C, Mulka JS, Hajela N (2015) Sustained fitness gains and variability in fitness trajectories in the long-term evolution experiment with *Escherichia coli*. *Proc Biol Sci* 282:20152292. <https://doi.org/10.1098/rspb.2015.2292>
9. Lenski RE (2017) Experimental evolution and the dynamics of adaptation and genome evolution in microbial populations. *ISME J* 11:2181–2194. <https://doi.org/10.1038/ismej.2017.69>
10. Vinchhi R, Jena C, Matange N (2023) Adaptive laboratory evolution of antimicrobial resistance in bacteria for genetic and phenotypic analyses. *STAR Protoc* 4:102005. <https://doi.org/10.1016/j.xpro.2022.102005>
11. Sanger F, Nicklen S, Coulson AR (1977) DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci U S A* 74:5463–5467. <https://doi.org/10.1073/pnas.74.12.5463>
12. Shendure J, Ji H (2008) Next-generation DNA sequencing. *Nat Biotechnol* 26:1135–1145. <https://doi.org/10.1038/nbt1486>
13. Schwengers O, Jelonek L, Dieckmann MA, Beyvers S, Blom J, Goesmann A (2021) Bakta: rapid and standardized annotation of bacterial genomes via alignment-free sequence identification. *Microb Genom* 7:000685. <https://doi.org/10.1099/mgen.0.000685>

14. Chen S, Zhou Y, Chen Y, Gu J (2018) fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics* 34:i884–i890. <https://doi.org/10.1093/bioinformatics/bty560>
15. Gurevich A, Saveliev V, Vyahhi N, Tesler G (2013) QUASt: quality assessment tool for genome assemblies. *Bioinformatics* 29:1072–1075. <https://doi.org/10.1093/bioinformatics/btt086>
16. Seeman T (2018) Snippy: fast bacterial variant calling from NGS reads (<https://github.com/tseemann/snippy>)
17. Wick RR, Judd LM, Gorrie CL, Holt KE (2017) Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Comput Biol* 13:e1005595. <https://doi.org/10.1371/journal.pcbi.1005595>

Tables

Table 1. Software used to analyse WGS data.

Name	Description	Reference
Bakta	Annotate bacterial genomes from genome files in fasta format. It uses a database implemented in the software to annotate.	[13]
fastp	Trim bad-quality reads and eliminate the completely bad ones. It uses raw fastq files as input.	[14]
Quast	Assess the quality of a de novo assembly.	[15]
snippy	Variant calling of bacterial strains using a bacterial reference genome.	[16]
Unicycler	Assemble a de novo genome from reads.	[17]

Table 2. Mutations in resistant variants of *Escherichia marmotae* Ec219 (snp: single nucleotide polyphormism ; del: deletion)

Variant	Contig	Position	Type	Reference	Alternate	Effect	Gene
Ec219_W3_Ri f100R_H2O	7	284902	del	TG	T	Frameshift variant	<i>rbsR</i>
Ec219_W3_Ri f100R_H2O	22	5608	snp	C	T	Missense variant	<i>rpoB</i>
Ec219_W4_Ri f20R_TMN	2	428187	snp	G	C	Missense variant	<i>rpoD</i>
Ec219_W4_Ri f20R_TMN	13	5453	snp	C	T	-	-

Figure 1. Analysis pipeline of whole genome sequencing data

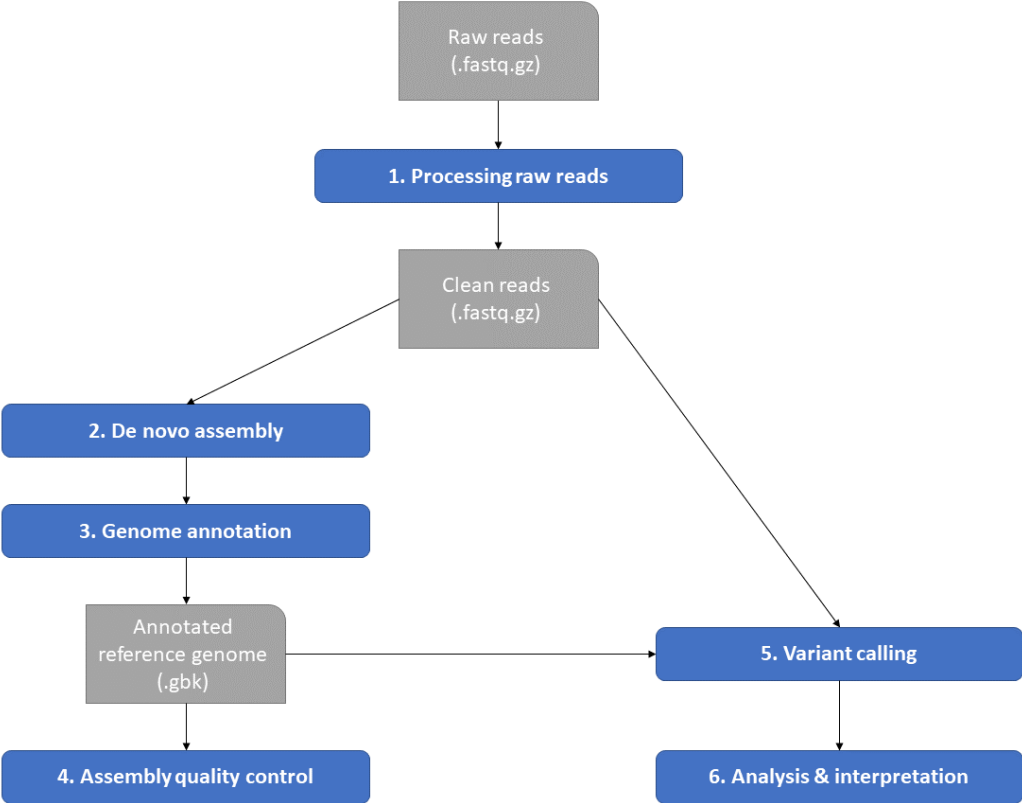


Figure 2. Comparison of *rbsR*, *rpoB* and *rpoD* DNA sequences between the Ec219 parental strain and the two variants Ec219_W3_Rif100R_H2O and Ec219_W4_Rif20R_TMN.

