Copy Number Variations

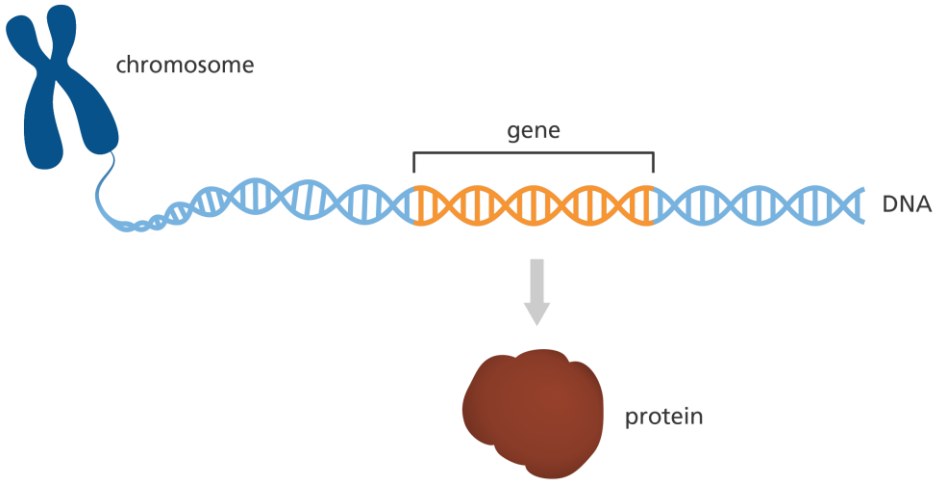
UNIVERSITÀ DEGLI STUDI DELL'AQUILA
DISIM
Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Andrea D'Angelo / Contact: andrea.dangelo6@graduate.univaq.it

Università degli Studi dell'Aquila / Italy

Genes

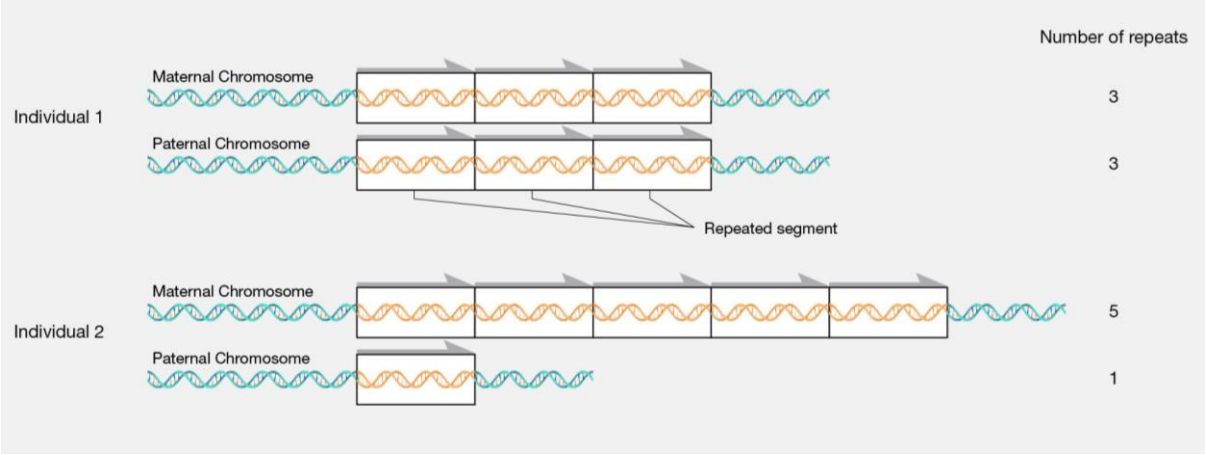
- A **gene** is a small section of the DNA that codes for a protein.



- Each gene contains the instructions for making a specific protein.

What are Copy Number Variations?

- A Copy Number Variation (**CNV**) is a difference in the number of copies of a gene or region between individuals.



- A CNV is a type of genetic variation where some sections of the genome are repeated or deleted.

What are Copy Number Variations? (2)

- Normally, individuals inherit 2 copies of each gene, one for each parent, resulting in a diploid genome.

However, it may happen that an individual has 0, 1, or more than 2 copies of each gene. That is a **CNV**.

Deletions and Duplications

- If an individual has 0 or 1 copies of a gene, we talk about a **deletion CNV**.
- If there are more than 2 copies, we talk about a **duplication CNV**.
- CNVs contribute to the human population diversity and evolutionary processes. However, they can also be biomarkers for certain pathological processes such as cancer.

Difference with other variations

- A Single Nucleotide Variant (SNV) is the simplest form of genetic variation. It occurs when a single nucleotide is altered.

- Indels are mutations where small pieces of DNA are either inserted or deleted from the genome. They have a larger impact than SNV because they can change the reading frame of a gene (frameshift).

- CNVs are a type of variations where large segments of DNA have varying copy numbers. These segments can be amplified or deleted.

Importance of CNVs

CNVs are important because:

- They are a key factor in human diversity and evolution.
- CNVs are responsible for susceptibility to diseases or disorders, like autoimmune diseases.
- CNVs can affect gene expressions and, consequently, the functions of the cell.

CNV Detection Tools

Some of the most common tools for CNV detection include:

- Cn.MOPS uses a Bayesan approach to decompose read variations
- ExomeDepth compares read depths against a reference set to find differences.
- cnvGSA is useful to interpret CNVs in the context of biological pathways and processes.

- There are many more that employ a variety of different approaches.

Task

The segments of DNA that are affected by CNVs can range from a thousand to several million base pairs.

For this reason, locating and analyzing CNVs in a genome is a complex task.

The genome files need to be preprocessed and mapped before we attempt to use a CNV detection tool.

Bioinformatics framing

When defining a bioinformatics task, we usually define a pipeline of operations that we want to perform on our dataset.

In this case, our task will be:

CNV Detection Task:

Given a reference genome and a set of DNA samples, locate possibly pathogenic CNV in the samples.

Bioinformatics framing (2)

CNV Detection Task:

Given a reference genome and a set of DNA samples, locate possibly pathogenic CNV in the samples.

The reference genome is a baseline sequence, for instance, the typical sequence of the human genome.

Bioinformatics framing (3)

CNV Detection Task:

Given a reference genome and a set of DNA samples, locate possibly pathogenic CNV in the samples.

The DNA samples are a set of samples from the same family as the reference genome.

If we have human DNA samples → We will use the human reference genome

Bioinformatics framing (4)

CNV Detection Task:

Given a reference genome and a set of DNA samples, locate possibly pathogenic CNV in the samples.

Suppose we located some CNVs. How do we know if they are possibly pathogenic?

Through a process called **Annotation** (more on that later), we compare the CNV we found with databases containing CNVs that were already found and analyzed in real laboratories.

Pipeline

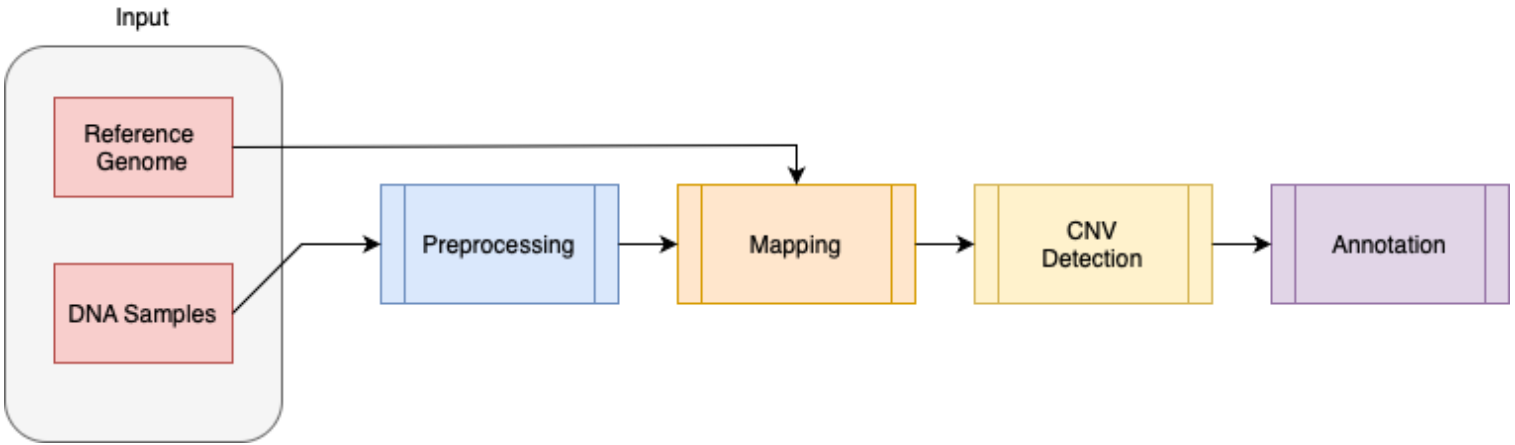
Whenever we define a bioinformatics task, we also want to define a pipeline of steps to solve it.

Each step is a data processing operation on the input samples aimed at obtaining the desired output.

For our CNV Detection task, we know we need to preprocess and map the samples, and then detect and annotate the CNVs.

Pipeline (2)

This is a high-level overview of the pipeline to solve our task.



We will go through each step, one by one.

Input

The reference genome is usually available online for download.

It must match the species of the DNA Samples we want to investigate (e.g., human, escherichia coli, etc…). DNA Samples are often referred to as «reads».

For this example, we assume that the DNA Samples are given in the FASTQ file format.

Input (2)

FASTQ Files are text files written as:

… CCGTAGCGAATGCGTATGCA …

+

… AAAAAAAA::99@@A??FCAAA …

Sequence of DNA bases.
You only need to specify one because the other one is given.
(e.g. A can only link with T)

Quality scores as ASCII characters

Input (3)

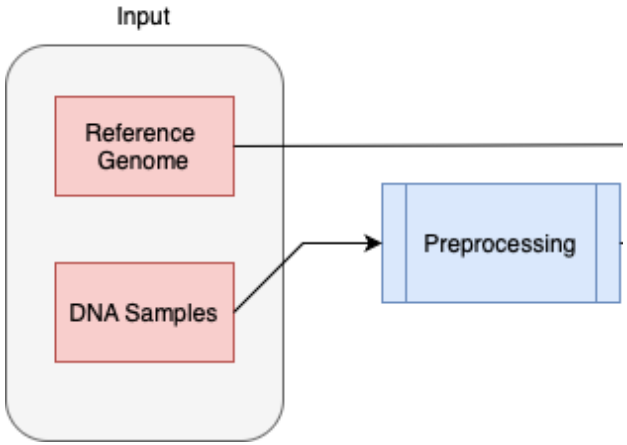
The reference genomes are usually given as FASTA files, which only contain the first row (the sequence), without the quality attributes.

Reference genomes are **usually** given in FASTA format.

DNA Samples are **usually** given in FASTQ format.

FASTQ Files are very simple but they are huge in size. Just one human DNA Sample can weight 5 Gb.

Preprocessing



The Preprocessing step of the pipeline involves trimming the samples and evaluating their quality.

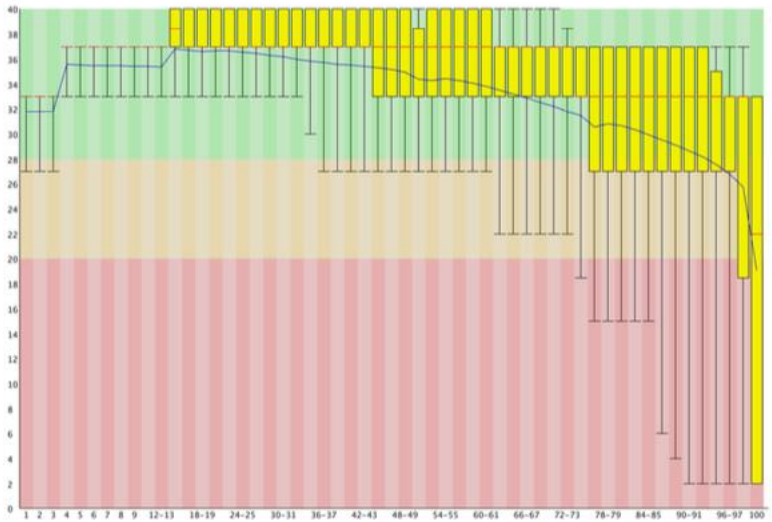
Trimming is the process of removing low-quality or unwanted sequences from the DNA samples.
It can be done if the samples are in the FASTQ format.

Evaluating the quality of a read means assessing its reliability and accuracy. It depends on multiple factors.

Usually reads from trusted sources have already been evaluated.

Preprocessing (2)

Example of the impact of trimming on read quality.



High quality

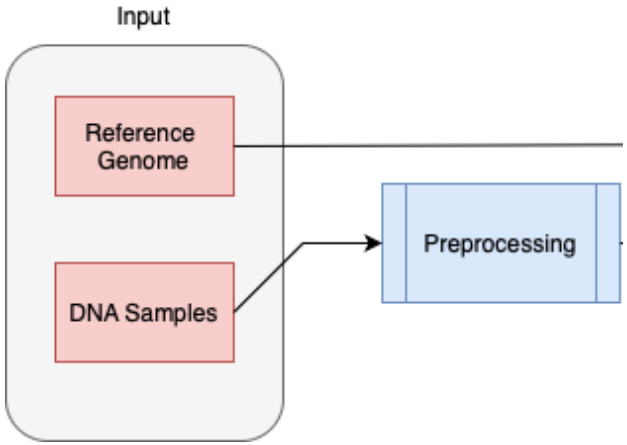
Low quality

Before trimming

After trimming

Trimming removed the reads with the lowest quality.

Preprocessing (3)



The Preprocessing step of the pipeline involves trimming the samples and evaluating their quality.

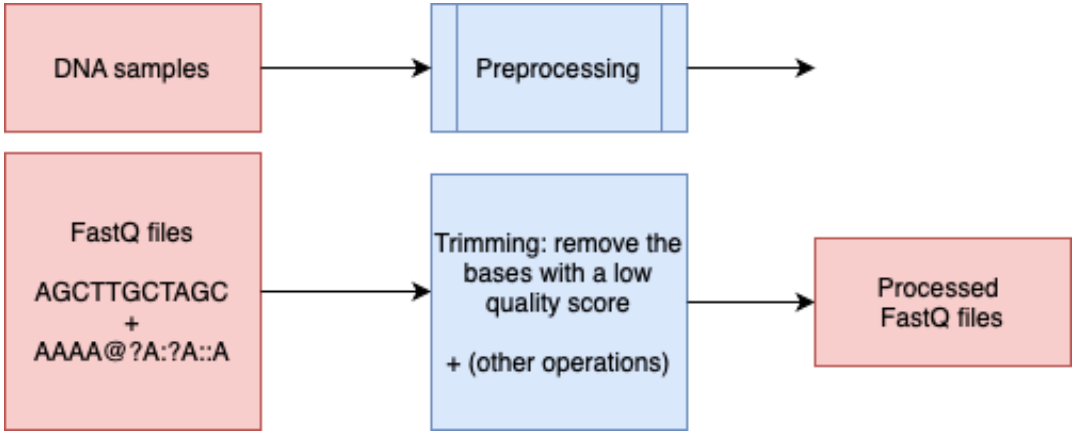
The Preprocessing step only takes the DNA samples as input.

It does not need the Reference Genome.

The output of the Preprocessing step are files in the same format as the input, possibly trimmed or processed in some other way.

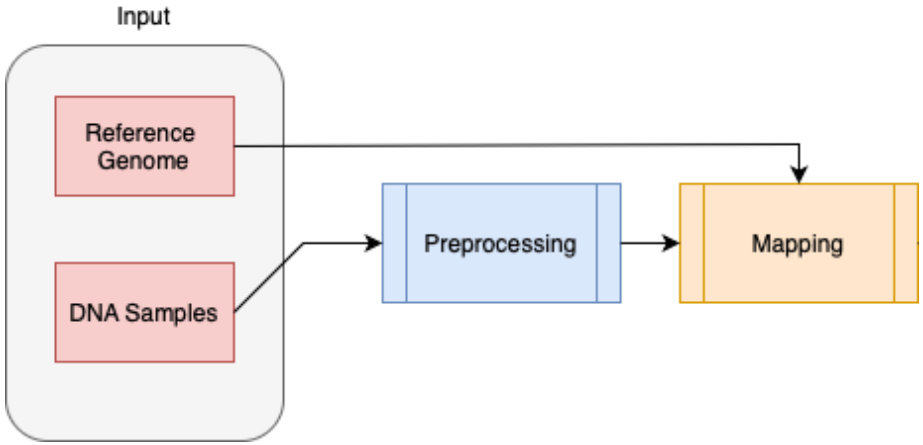
Preprocessing (4)

The detailed pre-processing looks like this:



Other preprocessing operations could be Quality Control, Adapter Removal, Normalization, etc…

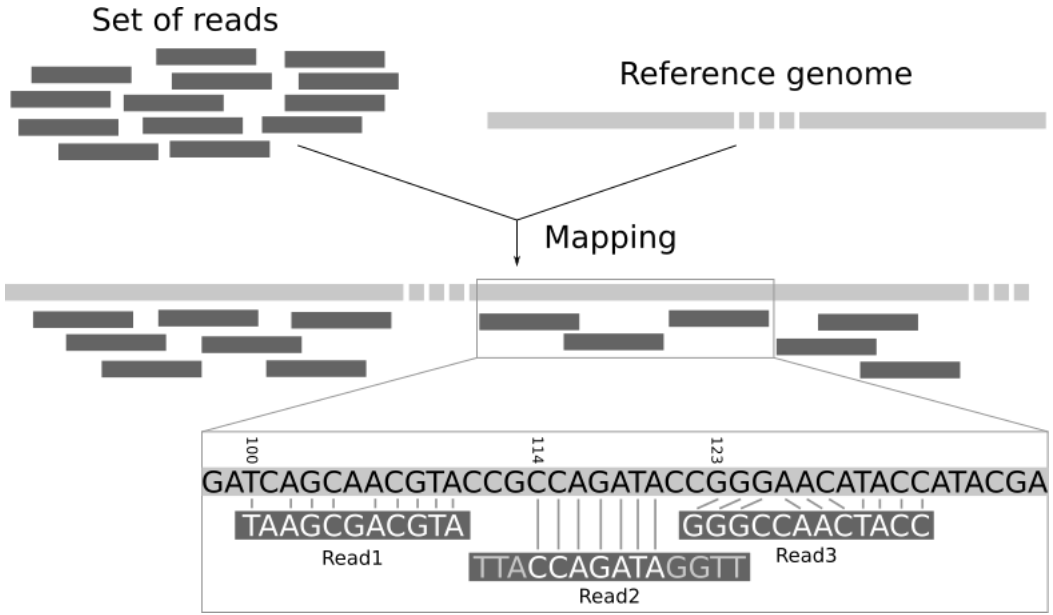
Mapping



Mapping is the process of aligning sequencing reads from a sample to a reference genome.

We are trying to find where each read fits inside of the reference genome.

Mapping (2)



Each read is mapped to a specific location of the Reference genome.
This is crucial for future analysis.

Mapping (3)

However, reads will rarely fully align 1:1 with the reference genome.

There might be variations, indels, SNVs, CNVs, to take into consideration.

The mapping process tries to find the best possible place for all sample reads. This way, we know what region we are working on with the highest probability.

Mapping (4)

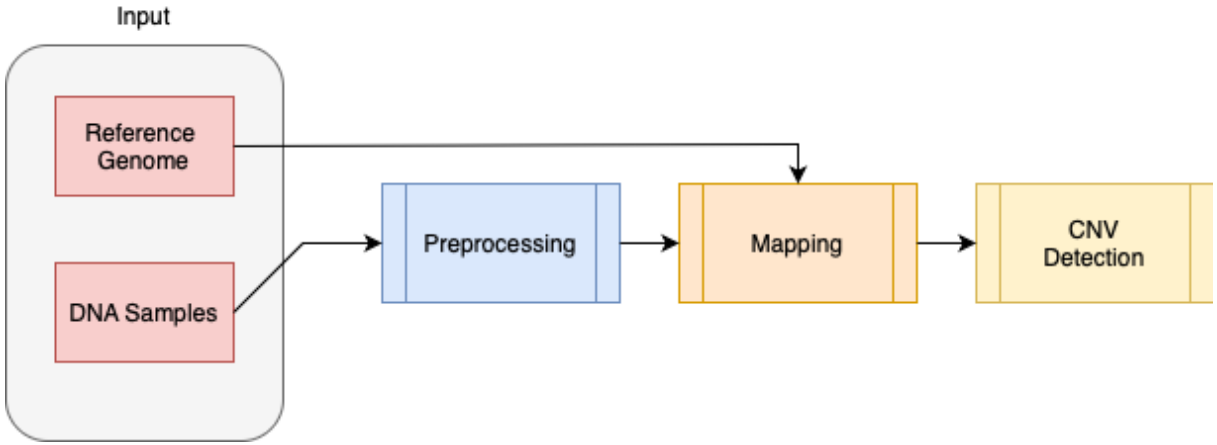
The output of a Mapping procedure is a BAM file.

BAM stands for Binary Alignment/Map format.

A BAM File is a binary file that stores the aligned sequences reads in relation to the reference genome.

It contains detailed information about each read and how it aligns to the reference genome.

CNV Calling



The process of CNV detection is often referred to as «CNV calling».

There are many different tools for CNV detection, and they all use different approaches.

CNV Calling (2)

For instance, the tool called **cn.mops** uses these steps (remember that each sample can be made of multiple reads):

1. It starts by counting how many reads map to various regions of the genome. Regions with more reads might indicate extra copies of that DNA segment (**Duplicate CNVs**), while regions with no reads might indicate **Deletion CNVs**.
2. However, since this might be due to various other reasons, cn.mops uses statistical models (in particular a mixture of Poisson distributions) to model these counts and find outliers.
3. It then compares the variations to other samples, making it easier to spot true CNVs.

CNV Calling (3)

The output of a CNV calling tool will include detailed informations about the CNVs that were found in each sample, including:

- Its location with respect to the reference genome
- Whether it is a duplication or a deletion
- The size of the sequence involved
- The quality score or confidence of the calling

CNV Calling (4)

For instance, one generic cn.mops output could be:

seqnames	start	end	width	sample	median	CN
chr1	10332971	10339296	6326	ERR166302.bam	-3.14°-6	CN1

Each row represents a CNV that the tool found.
In this example, we only analyze one CNV.

CNV Calling (5)

For instance, one generic cn.mops output could be:

seqnames	start	end	width	sample	median	CN
chr1	10332971	10339296	6326	ERR166302.bam	-3.14°-6	CN1

seqnames is the chromosome in which the CNV was found.

start and **end** are the initial and final positions in the reference genome of the affected sequence.

width measures the size of the sequence involved.

sample indicates the sample in which the CNV was found.

CNV Calling (6)

For instance, one generic cn.mops output could be:

seqnames	start	end	width	sample	median	CN
chr1	10332971	10339296	6326	ERR166302.bam	-3.14°-6	CN1

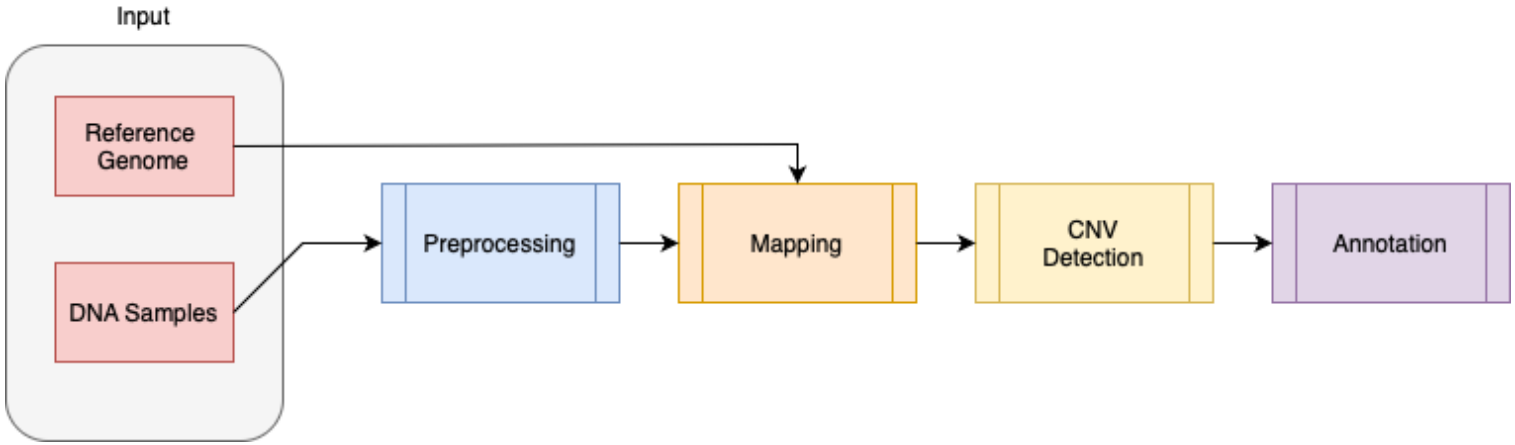
median is the median quality score associated with the CNV.

Simply put, it measures the reliability of the calling of this specific CNV.

Many quality measures can possibly be employed.

Lastly, **CN** tells us how many times the sequence was repeated. In this case, it's just one, which means this is a Deletion CNV (less than 2).

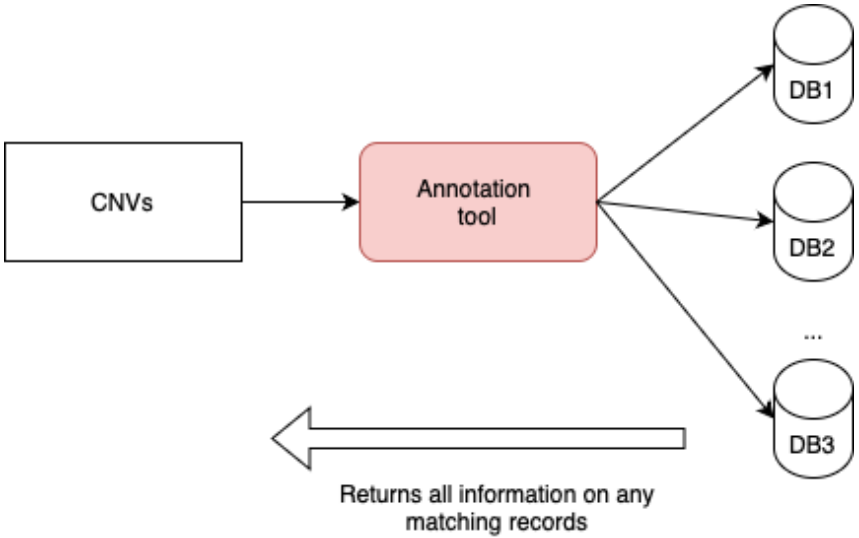
Annotation



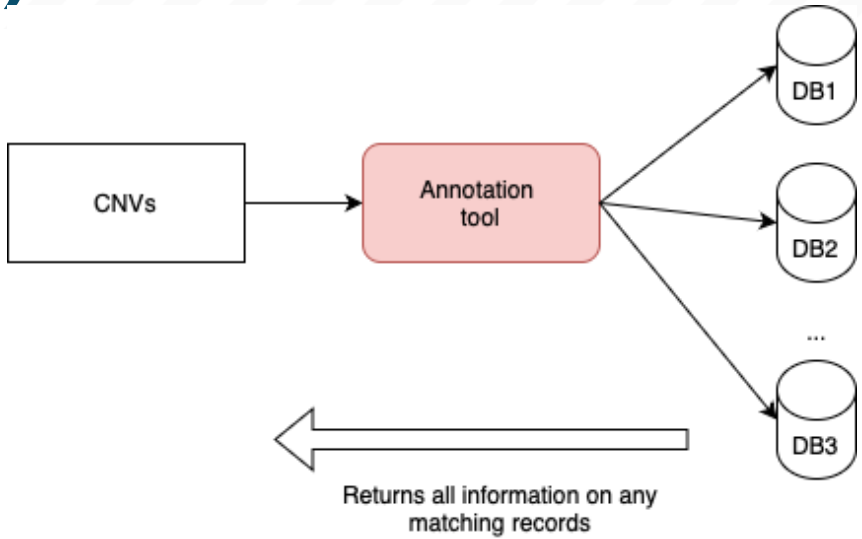
Now that we found all the CNVs in our samples, we need to understand whether they are dangerous or not.

Annotation (2)

To perform annotation, we use a tool that queries multiple Databases to check if any of the CNVs we retrieved in the previous phase match with any stored record.



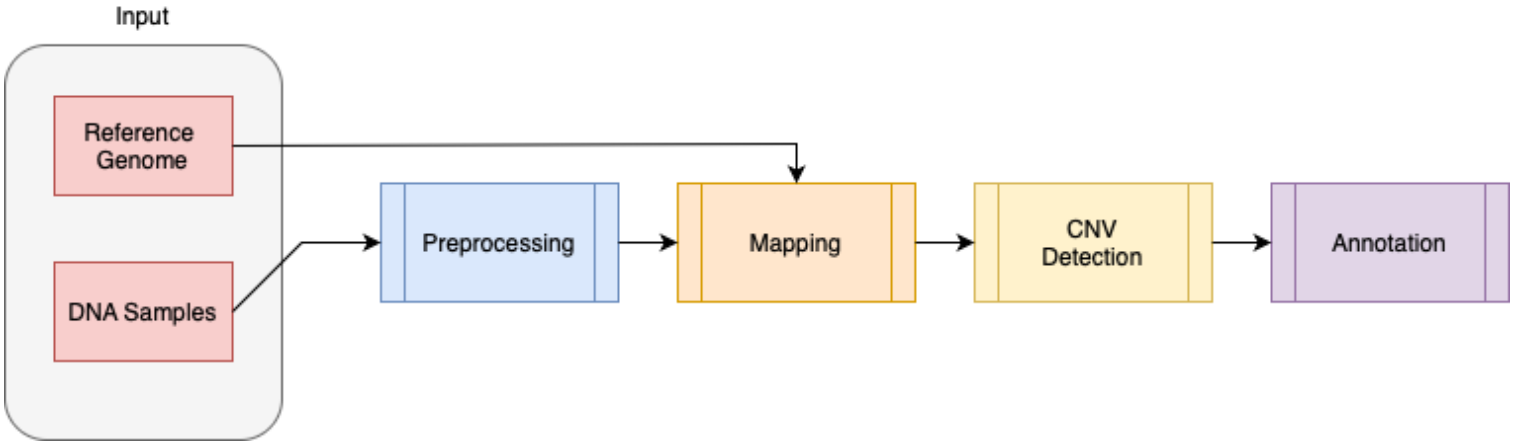
Annotation (3)



The resulting file contains all our CNVs annotated with all the possible information already verified in real laboratories. We have successfully fulfilled our task.

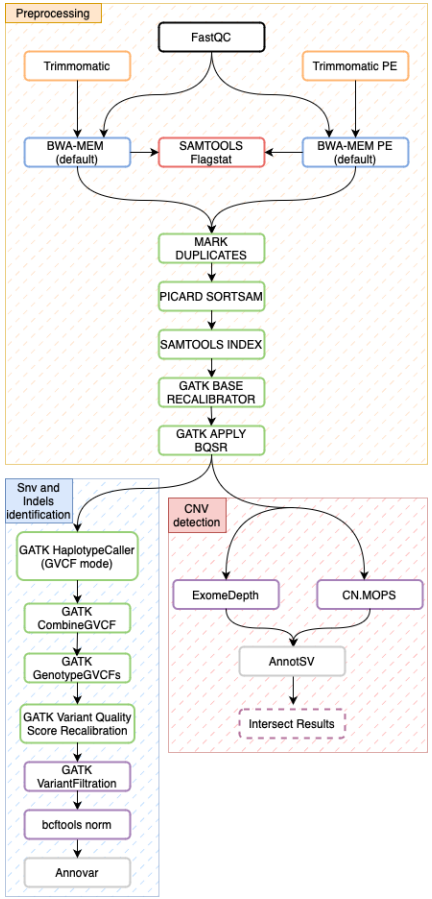
Pipeline

This is a high-level overview of the pipeline we defined.



In real-world applications, the pipeline follows the same general structure, but might be intrinsically more complex.

Pipeline (2)

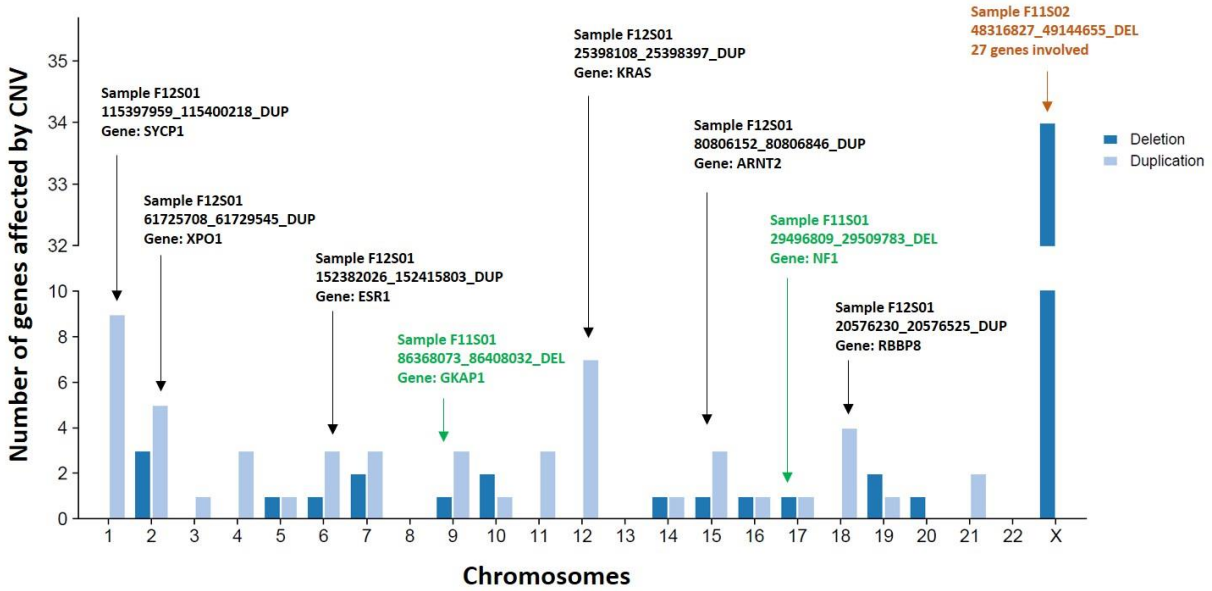


This is an example of a pipeline applied on real-world data of Breast cancer patients for the detection of SNVs, Indels, and CNVs.

The pipeline still follows the same general rules, despite being far more complex inside each individual step.

In this case, the pipeline uses two different CNV calling tools and intersect their results for reliability.

Pipeline (3)



These are some of the results we obtained with that pipeline. Before annotation, we already knew where each CNV resided, in terms of chromosomes. After annotation, we were also able to identify each gene involved.

References

1. https://training.galaxyproject.org/training-material/topics/sequence-analysis/tutorials/mapping/tutorial.html

2. https://bioconductor.org/packages/release/bioc/html/cn.mops.html