

# ADB-02-ngs-linux (hands-on)

---

## Transcriptome Assembly: Case study of bacteria *Listeria monocytogenes*

Fernando Pozo Wednesday, 2nd September, 2020

---

## Table of Contents

- [Transcriptome Assembly: Case study of bacteria \*Listeria monocytogenes\*](#)
    - [1. Setting up our working environment](#)
      - [1.1. Organism of interest: \*Listeria monocytogenes\*](#)
      - [1.2. Sequencing data: How to get it](#)
      - [1.3. Downloading our samples](#)
    - [2. How to run this tutorial](#)
      - [2.1. First steps with Docker](#)
      - [2.2. Configuring our Docker container](#)
      - [2.3. Software Requisites](#)
    - [3. Data preprocessing](#)
      - [3.1. Quality control with FastQC](#)
      - [3.2. Analysis of FastQC results](#)
      - [3.3. Trimming reads with Trimmomatic](#)
    - [4. Read mapping to a reference genome](#)
      - [4.1. Build an index and read mapping with Bowtie 2](#)
      - [4.2. Converting and sorting reads with Samtools](#)
    - [5. Transcriptome Assembly with Stringtie](#)
- 

### Note

All the links are [clickable](#). They contain references, databases and interesting sources to know more about our pipeline.

---

## 1. Setting up our working environment

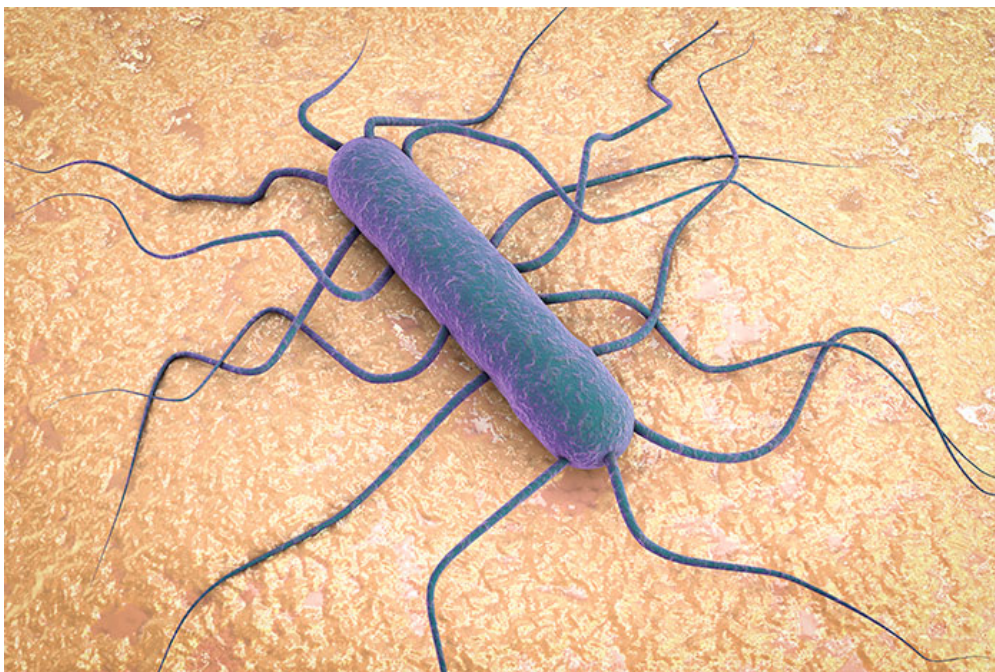
**DNA sequencing** is the process of determining the nucleic acid sequence – the order of nucleotides in DNA. It includes any method or technology that is used to determine the order of the four bases: adenine, guanine, cytosine, and thymine. The advent of rapid DNA sequencing methods has greatly accelerated biological and medical research and discovery.

The massively parallel sequencing technology known as **next-generation sequencing (NGS)** has revolutionized the biological sciences. With its ultra-high throughput, scalability, and speed, NGS enables researchers to perform a wide variety of applications and study biological systems at a level never before possible.

### 1.1. Organism of interest: *Listeria monocytogenes*

For this tutorial, we are going to use data of RNA sequencing from bacteria *Listeria monocytogenes*, a food-borne pathogen responsible for listeriosis, a sickness with a high mortality rate, which recently leads an [outbreak](#) in Spain.

The genome sequences of several *Listeria* species have opened a new era in the study of *Listeria*. These postgenomic studies, comparing pathogenic strains of *L. monocytogenes*, have led to the discovery of new virulence factors. Postgenomic studies have also helped us to understand how *Listeria* survives in a variety of different environments.



*L. monocytogenes* is one of the most virulent foodborne pathogens

## 1.2. Sequencing data: How to get it

We are going to access **RNA-seq** data from the project called "**Transcriptomic differences between virulence-attenuated and fully-virulent strains of *Listeria monocytogenes***", published some weeks ago (2019-07-16), that aims to compare the transcriptomic differences between virulence-attenuated and fully-virulent strains of *L. monocytogenes* under simulated food processing facility conditions. the study ***Listeria monocytogenes* sequencing under simulated food processing facility conditions.**, which is publicly available in SRA (a bioinformatics database that provides a public repository for DNA sequencing data). [SRA reference SRR9691134](#).

Our RNA sequencing (RNA-seq) data is stored in a pair of .fastq files (paired-end RNA-seq data [check differences between Single-end vs. paired-end reading](#)): these files are composed of millions of reads which were run on the [Illumina MiSeq](#) platform.

We will take as reference the [Listeria monocytogenes EGD-e whole genome](#). As *L. monocytogenes* is a well-studied organism, we will assume that the reference genome and assembly are mostly correct and complete.

It is also important to select a [FASTA sequence](#) and a [GFF annotation file](#), from the reference genome of our organism. Available in [Genome](#).

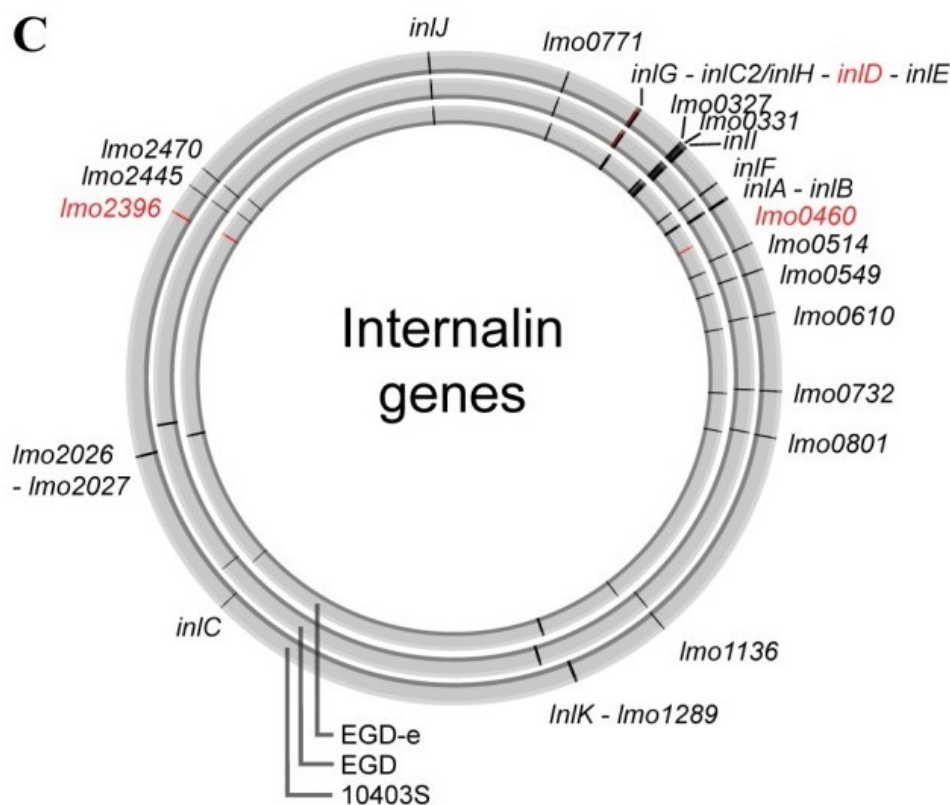


Figure from [Source](#).

### 1.3. Downloading our samples

First, please open a terminal session (ALT+t) and create a directory to put the samples and store the results, and giving permissions (r:read, w:write, x:execute):

```
mkdir -p ~/SAMPLES
chmod a+rwX ~/SAMPLES
mkdir -p ~/RESULTS
chmod a+rwX ~/RESULTS
```

We can download the RNA-seq data [here](#) or typing in a terminal:

```
wget -P ~/SAMPLES https://sra-pub-src-
1.s3.amazonaws.com/SRR9691134/V1701FE3T_S6_L001_R1_001.fastq.gz.1
wget -P ~/SAMPLES https://sra-pub-src-
1.s3.amazonaws.com/SRR9691134/V1701FE3T_S6_L001_R2_001.fastq.gz.1
```

and both FASTA sequence and GFF annotation file with:

```
wget -P ~/SAMPLES
ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/196/035/GCF_000196035.1_ASM1
9603v1/GCF_000196035.1_ASM19603v1_genomic.gff.gz
```

```
wget -P ~/SAMPLES
ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/196/035/GCF_000196035.1_ASM19603v1/GCF_000196035.1_ASM19603v1_genomic.fna.gz
```

### uncompressing:

```
gunzip ~/SAMPLES/GCF_000196035.1_ASM19603v1_genomic.gff.gz
gunzip ~/SAMPLES/GCF_000196035.1_ASM19603v1_genomic.fna.gz
```

### and finally renaming files:

```
mv ~/SAMPLES/GCF_000196035.1_ASM19603v1_genomic.fna
~/SAMPLES/lmonocytogenes_genome.fasta
mv ~/SAMPLES/GCF_000196035.1_ASM19603v1_genomic.gff
~/SAMPLES/lmonocytogenes_genome.gff
mv ~/SAMPLES/V1701FE3T_S6_L001_R1_001.fastq.gz.1
~/SAMPLES/lmonocytogenes_read1.fastq.gz
mv ~/SAMPLES/V1701FE3T_S6_L001_R2_001.fastq.gz.1
~/SAMPLES/lmonocytogenes_read2.fastq.gz
```

### Visualizing our downloaded samples we should obtain:

```
(base) fpozoc@jonflash:~/SAMPLES$ head lmonocytogenes_genome.fasta
>NC_003210.1 listeria monocytogenes EGD-e chromosome, complete genome
TAATTAATCTAACAAATTCGTTACAGATTTCTTTACACACAAGTTATACACAAGTTAACTGGCTGTGGACAACCGTTTT
TCACATCTGGACAGTTTTGTGGATAGATTGGTAAGTCCTTGCTATCAGAGTGATTTCTGATATTATAATTCCTGTGCGA
ATAGAAATATAGCTGGGGAAAACTAAAGTTATCCACAATACATTTTACTTTGTGGATAATTTTTTAACAGCTTTTGGAT
AACCTTATCCATAGCTTTTCTATCTGGGTGAACCTTATAGCATCCATTTACATTACATAAAAAAGGGGGGCTACTAGTG
CAATCAATTGAAGACATCTGGCAGGAAACACTGCAAAATTGTTAAAAAAAATATGAGTAAACCTAGTTACGATACATGGAT
GAAATCAACAACCGCTCATTCACTTGAAGGTAACACGTTTATTATTCAGCGCCCAATAATTTTGTCCGCGATTGGTTAG
AGAAAAGCTACACTCAATTTATCGCTAACATTTTGCAGAAATAACTGGTCGCTTATTTGATGTCGCGTTTATTGATGGC
GAGCAGGAAGAAGAACTTTGAATACACTGTGATTAACCAAATCCAGCATTAGATGAAGATGGCATTGAATTTGAAAACA
TATGCTTAAATCCACGTTATGTTTTGATACCTTTGTACTTGTTCAGGGAACAGATTTCGCCACGCGACATCACTTCGACG
```

### Output of head ~/SAMPLES/lmonocytogenes\_genome.fasta

```
(base) fpozoc@jonflash:~/SAMPLES$ head -n 30 lmonocytogenes_genome.gff
##gff-version 3
##gff-spec-version 1.21
##processor NCBI annotwriter
##genome-build ASM19603v1
##genome-build-accession NCBI Assembly:GCF_000196035.1
##sequence-region NC_003210.1 1 2944528
##species https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?tid=169963
NC_003210.1 RefSeq region 1 2944528 . + . ID=ld0;Dbxref=taxon:169963;Is_circular=true;Name=ANONYMOUS;gbkey=Src;genome=chromosome;mol_type=genomic DNA;strain=EGD-e
NC_003210.1 RefSeq ribosome_entry_site 305 310 . + . ID=ld1;gbkey=regulatory;regulatory_class=ribosome_binding_site
NC_003210.1 RefSeq operon 310 3012 . + . ID=ld2;expert=EXISTENCE:[PMID:19448609];gbkey=operon;operon=operon_001
NC_003210.1 RefSeq gene 318 1073 . + . ID=gene0;Dbxref=GeneID:984365;Name=dnaA;expert=EXISTENCE:[PMID:19448609];gbkey=Gene;gene=dnaA;gene_biotype=protein_coding;locus_tag=lnm0801
NC_003210.1 RefSeq CDS 318 1073 . + 0 ID=cds0;Parent=gene0;Dbxref=GeneBank:NP_463534.1, GeneID:984365;Name=NP_463534.1;Note=binds to the dnaA-box as an ATP-bound complex at the origin of replication during the initiation of chromosomal replication;38 me replication initiator DNA;protein_id=NP_463534.1;transl_table=11
NC_003210.1 RefSeq gene 1867 3012 . + . ID=gene1;Dbxref=GeneID:984379;Name=dnaN;expert=EXISTENCE:[PMID:19448609];gbkey=Gene;gene=dnaN;gene_biotype=protein_coding;locus_tag=lnm0802
NC_003210.1 RefSeq CDS 1867 3012 . + 0 ID=cds1;Parent=gene1;Dbxref=GeneBank:NP_463535.1, GeneID:984379;Name=NP_463535.1;Note=binds the polymerase to DNA and acts as a sliding clamp
expert=EXISTENCE:[PMID:19448609];gbkey=CDS;gene=dnaN;product=DNA polymerase III subunit beta;protein_id=NP_463535.1;transl_table=11
NC_003210.1 RefSeq ribosome_entry_site 1856 1860 . + . ID=ld3;gbkey=regulatory;regulatory_class=ribosome_binding_site
NC_003210.1 RefSeq terminator 3030 3062 . + . ID=td4;gbkey=regulatory;regulatory_class=terminator
NC_003210.1 RefSeq ribosome_entry_site 3104 3109 . + . ID=ld5;gbkey=regulatory;regulatory_class=ribosome_binding_site
NC_003210.1 RefSeq gene 3121 4464 . + . ID=gene2;Dbxref=GeneID:984420;Name=lnm0803;gbkey=Gene;gene=lnm0803
NC_003210.1 RefSeq CDS 3121 4464 . + 0 ID=cds2;Parent=gene2;Dbxref=GeneBank:NP_463536.1, GeneID:984420;Name=NP_463536.1;expert=EXISTENCE:[PMID:19448609];gbkey=CDS;product=hypothetical protein;protein_id=NP_463536.1;transl_table=11
NC_003210.1 RefSeq ribosome_entry_site 4631 4637 . + . ID=ld6;gbkey=regulatory;regulatory_class=ribosome_binding_site
NC_003210.1 RefSeq ribosome_entry_site 4856 4856 . + . ID=gene3;Dbxref=GeneID:984453;Name=lnm0804;expert=EXISTENCE:[PMID:19448609];gbkey=Gene;gene=lnm0804;locus_tag=lnm0804
NC_003210.1 RefSeq CDS 4644 4865 . + 0 ID=cds3;Parent=gene3;Dbxref=GeneBank:NP_463537.1, GeneID:984453;Name=NP_463537.1;expert=EXISTENCE:[PMID:19448609];gbkey=CDS;product=hypothetical protein;protein_id=NP_463537.1;transl_table=11
NC_003210.1 RefSeq CDS 4644 4865 . + 0 ID=ld7;Dbxref=GeneID:984453;gbkey=regulatory;regulatory_class=ribosome_binding_site
NC_003210.1 RefSeq ribosome_entry_site 4856 4861 . + . ID=gene4;Dbxref=GeneID:984623;Name=recF;gbkey=Gene;gene=recF;gene_biotype=protein_coding;locus_tag=lnm0805
NC_003210.1 RefSeq gene 4869 5981 . + . ID=cds4;Parent=gene4;Dbxref=GeneBank:NP_463538.1, GeneID:984623;Name=NP_463538.1;Note=Required for DNA replication;38 binds preferentially to single-stranded 2C linear DNA;expert=EXISTENCE:[PMID:19448609];gbkey=CDS;gene=recF;product=recombination protein F;protein_id=NP_463538.1;transl_table=11
NC_003210.1 RefSeq CDS 6030 7970 . + . ID=ld8;gbkey=regulatory;regulatory_class=ribosome_binding_site
NC_003210.1 RefSeq CDS 6030 7970 . + 0 ID=gene5;Dbxref=GeneID:984708;Name=gyrB;gbkey=Gene;gene=gyrB;gene_biotype=protein_coding;locus_tag=lnm0806
NC_003210.1 RefSeq subunit F protein;protein_id=NP_463539.1;transl_table=11 ID=cds5;Parent=gene5;Dbxref=GeneBank:NP_463539.1, GeneID:984708;Name=NP_463539.1;expert=EXISTENCE:[PMID:19448609];gbkey=CDS;gene=gyrB;product=DNA gyrase subunit F;protein_id=NP_463539.1;transl_table=11
NC_003210.1 RefSeq ribosome_entry_site 8048 8053 . + . ID=ld9;gbkey=regulatory;regulatory_class=ribosome_binding_site
NC_003210.1 RefSeq gene 8065 10593 . + . ID=gene6;Dbxref=GeneID:984791;Name=gyrA;gbkey=Gene;gene=gyrA;gene_biotype=protein_coding;locus_tag=lnm0807
```

### Output of head -n 10 ~/SAMPLES/lmonocytogenes\_genome.gff



```
(base) fpozoc@jonflash:~/SAMPLES$ zcat lmonocytoenes_read1.fastq.gz | head -n 10
@M000825:185:000000000-B5NDY:1:1101:17248:1026 1:N:0:6
NCCTTTTCCACCCAGCAGGAAATCGTCAGAACCTGCACGTTTTTCATTTCTGTAGGTTTTTTTTTTTTTCC
+
#8ACCFGGGFCFGGG7@F<<8,,@<FF;FF8;9<EFEE8E8,,CFFFF,CCEF,,,,,,,,,;,,+88@@>>,,
@M000825:185:000000000-B5NDY:1:1101:14316:1046 1:N:0:6
CTAACTGTAGCAAAACAACTTCAAACAGAGTTAGTTAAAGCCGTTTGAAGTGTTAATGACAATTACAAATTATT
+
8A6@BFDGD9EADCCFCFFFGGAA9F@<,CEF<@FF<9C9FF77FFGG8,,C,,,,,<,;,,,,,<,;,,,<,
@M000825:185:000000000-B5NDY:1:1101:17565:1052 1:N:0:6
CCAGAAATCGTTAATATCGAAACCAACCCAGCGCTTCTACGTGTGACATCACCGCTCTCCATTATTTCCCTTTTCC
+
8866-@-EE;FFC<EAFG@88,CFC@CEF>8FCEFEFG9F<F,C89F9CC,CE7++8,,,:<,,9,,<,999,:9,
@M000825:185:000000000-B5NDY:1:1101:10843:1053 1:N:0:6
CTAAAGTTACTTCTTCTGCTTCTATGGCAGATTTTATGGTGTTTCCGAACAACAAATCTTCTCCTTTCTTCT
+
8AA<@CFFFCFGGGGGGFEFGGCF?E8;8FEFGG?FCDF,FGF@E7,,@7,6,,,;,,,6,6,,,;,,,,,
@M000825:185:000000000-B5NDY:1:1101:12313:1064 1:N:0:6
AATGCCAAAATTAAGCTTACTTAAACAAGATGGAACAAACGCTGGCGAAATTACTTTAAACGACACTGTTTTCCGT
+
-8C<<EDFDDFEFFGGGGGFGGFFGGGGCCGGGFGDFEFGCGDCCF7++CF,CC<C,,C,+8+79,<C@CC,,:
@M000825:185:000000000-B5NDY:1:1101:15373:1067 1:N:0:6
CTCAACAAATGGACAAAGTCGGCGCAGACTTCTATACTCTGTAGTACTTTGCATTAACTTTTCTCCTCCC
+
-AA<CEFGCCFA9CE@<DCFE77F::++CFFFGG9F9FFGGAF9C<C,CFDF,,<,,,;<@,,:9,96,,6,
@M000825:185:000000000-B5NDY:1:1101:17535:1068 1:N:0:6
TTTAATGTAAGATTAGATGTACCCTAGTGAATCAAATGGATGCTCCGACGCTTTTAAACGGCTGTGAAGTAACA
+
6AC@CGGGGFFGGGGGGGGGGGGGGGGGEGGFCGGFFDGGGGGFGGGG7:F:CFGEEF8,C,+8@+:,,,<,,<,
@M000825:185:000000000-B5NDY:1:1101:16417:1080 1:N:0:6
CCCAAAACAATCATTGGCGAAGATAATACGCTTAGAACTCTCCAAATACTTGGACACCTGTGGAATGCCATTT
```

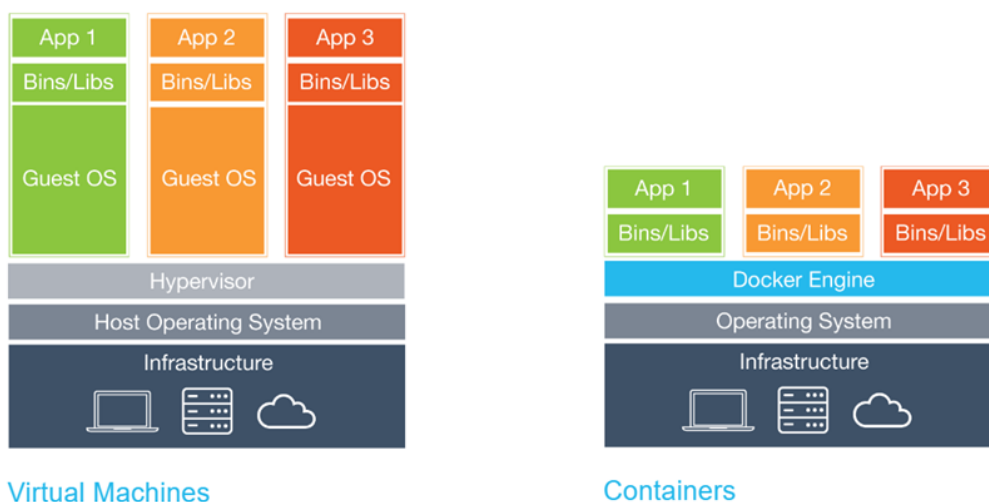
Output of `zcat ~/SAMPLES/lmonocytoenes_read1.fastq.gz head -n 10`

## 2. How to run this tutorial

Once we have downloaded and stored in `~/SAMPLES/` our FASTQ R1 and R2 files, the FASTA whole genome reference sequence and GFF annotation file, we can run our analysis inside a Docker container.

### 2.1. First steps with Docker

**Docker** is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.



All bioinformatics packages required to run the pipeline will be inside our Docker container

To use this tutorial, we are going to use an already constructed Docker image. This image contains inside the software required to properly run the analysis.

Please, **open a Linux terminal pressing ALT + t** and type:

```
docker -v
```

to check if you have installed Docker in your Linux OS.

Otherwise, you can visit the [Docker installation guide](#) to install it locally in your computer. *Remember that if you don't have administrator privileges you won't be able to install Docker.*

The image we are going to use can be pulled from [dockerhub](#). To get the last image to create our Docker container type in terminal and run:

```
docker pull osvoldogc/ufv:2.0
```

Once it has finished, you can list all top level images, their repository and tags, and their size with:

```
docker images
```

## 2.2. Configuring our Docker container

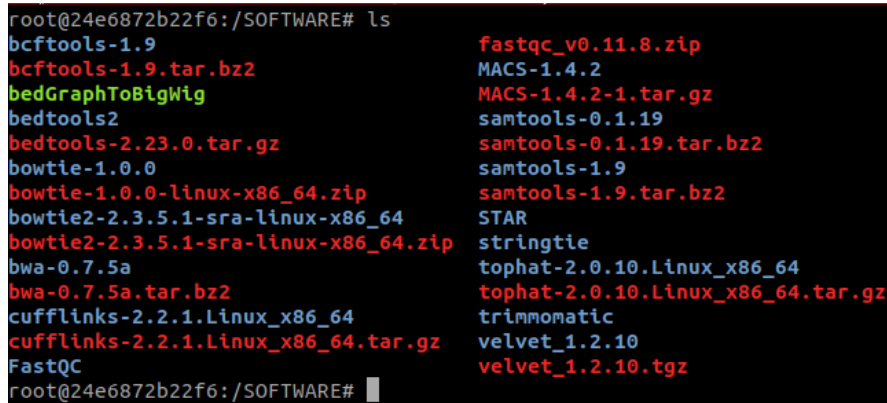
With our image already stored in our computer, we are going to declare our variables to define where will be our local and docker folders.

```
SAMPLES_LOCAL=~ /SAMPLES  
SAMPLES_DOCKER=/SAMPLES  
RESULTS_LOCAL=~ /RESULTS  
RESULTS_DOCKER=/RESULTS
```

And finally running the process in our isolated container:

```
docker run --rm -v $SAMPLES_LOCAL:$SAMPLES_DOCKER -v  
$RESULTS_LOCAL:$RESULTS_DOCKER -it osvoldogc/ufv:2.0 /bin/bash
```

*NOTE: with -v argument you are going to be able to interact locally with your Docker results in ~/SAMPLES and ~/RESULTS.*



```
root@24e6872b22f6:/SOFTWARE# ls
bcftools-1.9
bcftools-1.9.tar.bz2
bedGraphToBigWig
bedtools2
bedtools-2.23.0.tar.gz
bowtie-1.0.0
bowtie-1.0.0-linux-x86_64.zip
bowtie2-2.3.5.1-sra-linux-x86_64
bowtie2-2.3.5.1-sra-linux-x86_64.zip
bwa-0.7.5a
bwa-0.7.5a.tar.bz2
cufflinks-2.2.1.Linux_x86_64
cufflinks-2.2.1.Linux_x86_64.tar.gz
FastQC
fastqc_v0.11.8.zip
MACS-1.4.2
MACS-1.4.2-1.tar.gz
samtools-0.1.19
samtools-0.1.19.tar.bz2
samtools-1.9
samtools-1.9.tar.bz2
STAR
stringtie
tophat-2.0.10.Linux_x86_64
tophat-2.0.10.Linux_x86_64.tar.gz
trimmomatic
velvet_1.2.10
velvet_1.2.10.tgz
root@24e6872b22f6:/SOFTWARE#
```

osvaldogc/ufv:2.0 initial screen

And change our path to root:

```
cd ..
```

## 2.3. Software Requisites

Remember that all the packages required to run the analysis are already installed in our [Docker image](#).

Follow this [guide](#) if you want to run your analysis locally.

## 3. Data preprocessing

Data from HTS platforms can contain **adapters** and a variety of **experimental artifacts** and **low-quality data**, so data preprocessing and quality control are necessary.

### 3.1. Quality control with FastQC

FastQC, written by Simon Andrews of Babraham Bioinformatics, is a program designed to spot potential problems in HTS datasets. It runs a set of analyses on one or more raw sequence files in fastq or bam format and produces a report which summarises the results.

Andrews S. (2010). *FastQC: a quality control tool for high throughput sequence data*. Bioinformatics.

Modern high throughput sequencers can generate tens of millions of sequences in a single run. Before analysing this sequence to draw biological conclusions you should always perform some simple quality control checks to ensure that the raw data looks good and there are no problems or biases in your data which may affect how you can usefully use it.

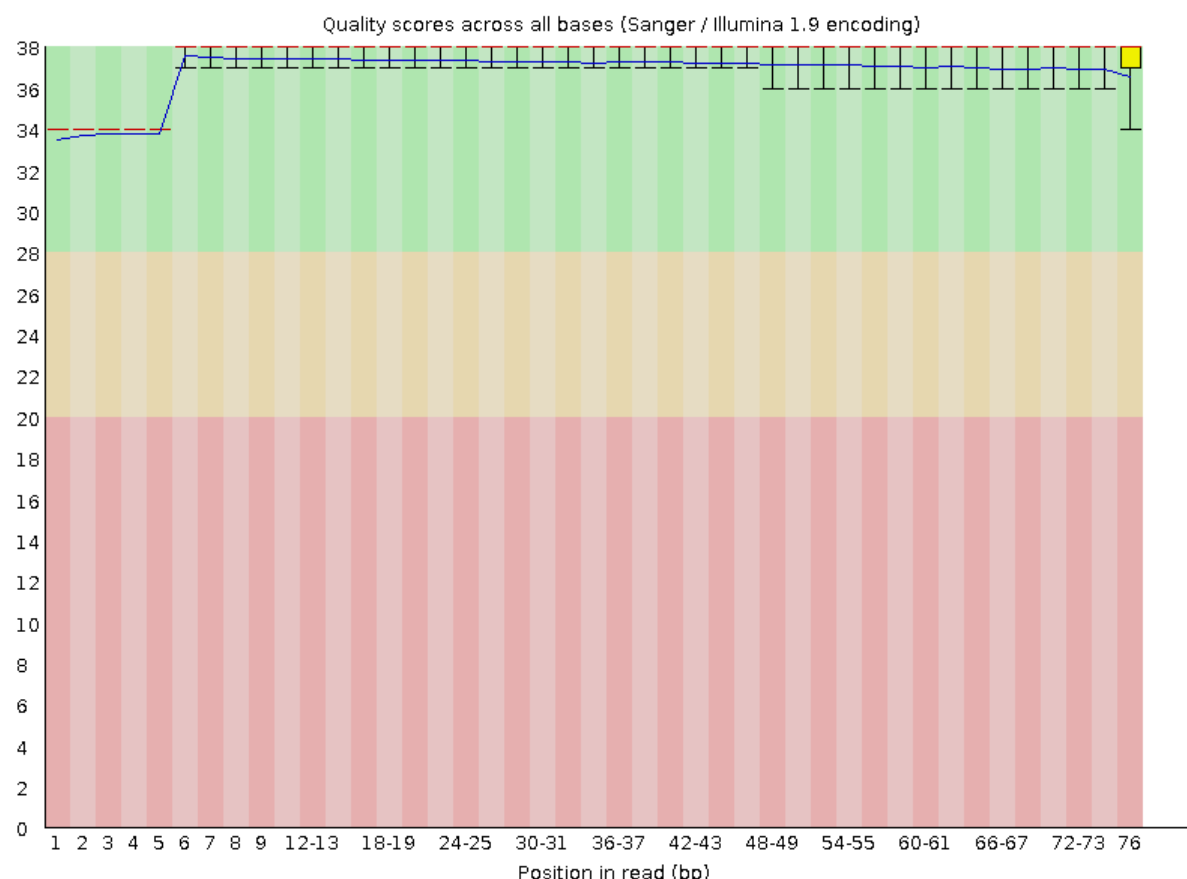
A report with basic statistics, sequence quality, sequence content scores, GC content scores, N content per base, duplication levels, overrepresented sequences, adapter content or kmer content will be given by the software.

To run the initial analysis, please run (*input, output and number of threads are the arguments below*):

```
fastqc /SAMPLES/lmonocytogenes_read1.fastq.gz -o /RESULTS -t 4
fastqc /SAMPLES/lmonocytogenes_read2.fastq.gz -o /RESULTS -t 4
```

### 3.2. Analysis of FastQC results

Report analysis could be completely covered following the [FastQC manual](#). Therefore, checking our html and zip files with images and reports is going to give us an idea about the quality of the sequencing.



*lmonocytogenes\_read1.fastq* quality per base

The most common types of quality control and procedures are:

- **Per-base quality:** Trim ends if quality too low.
- **Per sequence quality:** Average quality score should be well above 25-30, otherwise experiment may need to be repeated.
- **Sequence duplication levels:** A large number of identical sequence may indicate duplication by PCR during library preparation, this can bias estimates of mRNA expression and is often filtered out.
- **Overrepresented sequences:** Sometimes adapter sequences are represented in final sequences and these need to be filtered out before analysis.

This view shows an overview of the range of quality values across all bases at each position in the FastQ file. The quality seems to be enough to continue the analysis (the scope of this tutorial is not shown an extensive analysis of this step), but above we have shown that QC could be optimize following these procedures.



### 3.3. Trimming reads with Trimmomatic

Trimmomatic is an efficient preprocessing tool, which could correctly handle paired-end data, that can be used to trim and crop Illumina (FASTQ) data as well as to remove adapters.

Bolger AM, Lohse M, Usadel B. (2014) *Trimmomatic: a flexible trimmer for Illumina sequence data*. Bioinformatics.

NGS sequencing typically results in millions of reads. A proportion of these reads will contain **artifacts** or **low-quality** bases which we would like to **remove** before starting our analyses.

Trimmomatic performs a variety of useful trimming tasks for Illumina Paired-End (and also Single-End) data. The selection of trimming steps and their associated parameters are supplied on the command line.

*NOTE:* If you want to know which adaptors are in your data, the easiest is probably to ask lab person that generated the data. What you will usually observe with Illumina is indexed adaptor at the beginning of your F reads (plus sometimes some fake polyA stretches) and universal adaptor at the end of your R reads. I would recommend reading pages 5 and 6 in Trimmomatic [user guide](#) to know more about which types of adaptors could be in your data.

As we don't know with detail how the experiment was performed, For this tutorial, we are going to use one indexed and one universal adaptor, only as a toy sample, in order to show how to trim adaptors with Trimmomatic.

To automatically create a file with the content below, please type in the terminal:

```
printf ">PrefixAdapter7/1
GATCGGAAGAGCACACGTCTGAACTCCAGTCACCAGATCATCTCGTATGCCGTCTTCTGCTTG
>PrefixUniversal/2
AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT"
> /SAMPLES/illumina_adapter.fasta
```

We are going to use the next arguments to run our analysis:

- ILLUMINACLIP: Cut adapter and other illumina-specific sequences from the read.
- SLIDINGWINDOW: Performs a sliding window trimming approach. It starts. Scanning at the 5' end and clips the read once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality-
- TRAILING: Cut bases off the end of a read, if below a threshold quality-
- MINLEN: Drop the read if it is below a specified length-

And run Trimmomatic with the command:

```
java -jar /SOFTWARE/trimmomatic/bin/trimmomatic.jar PE -phred33 \
/SAMPLES/lmonocytogenes_read1.fastq.gz \
/SAMPLES/lmonocytogenes_read2.fastq.gz \
/RESULTS/lmonocytogenes_read1_paired.fastq \
/RESULTS/lmonocytogenes_read1_unpaired.fastq \
/RESULTS/lmonocytogenes_read2_paired.fastq \
```

```
/RESULTS/lmonocytogenes_read2_unpaired.fastq \  
ILLUMINACLIP:/SAMPLES/illumina_adapter.fasta:2:33:20:2:true \  
LEADING:36 \  
TRAILING:32 \  
SLIDINGWINDOW:4:30 \  
MINLEN:3
```

Again, we can then run the FastQC analysis again to see if we have fixed some things we saw previously.

```
fastqc /RESULTS/lmonocytogenes_read1_paired.fastq -o /RESULTS -t 4  
fastqc /RESULTS/lmonocytogenes_read2_paired.fastq -o /RESULTS -t 4
```

## 4. Read mapping to a reference genome

After we check to make sure that our raw reads are OK, we are going to construct our index and map the reads to our reference genome with Bowtie2.

### 4.1. Build an index and read mapping with Bowtie 2

**Bowtie 2** is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning reads of about 50 up to 100s or 1,000s of characters. Bowtie 2 indexes the genome with an [FM Index](#). Bowtie2 supports gapped, local, and paired-end alignment modes.

Langmead B, Salzberg S. (2012) [Fast gapped-read alignment with Bowtie 2](#). Nature Methods

We create a Bowtie 2 index in order to map efficiently million of reads to the reference FASTA sequence:

```
bowtie2-build /SAMPLES/lmonocytogenes_genome.fasta  
/RESULTS/lmonocytogenes_genome
```

Once we have built the index, we have to tell Bowtie 2 where to find these index files:

```
export BOWTIE2_INDEXES=/RESULTS
```

And then, map our reads to *L. monocytogenes* reference, which result in a .SAM file.

```
bowtie2 -x /RESULTS/lmonocytogenes_genome -p 2 -1  
/RESULTS/lmonocytogenes_read1_paired.fastq -2  
/RESULTS/lmonocytogenes_read2_paired.fastq -S  
/RESULTS/lmonocytogenes_genome.sam
```

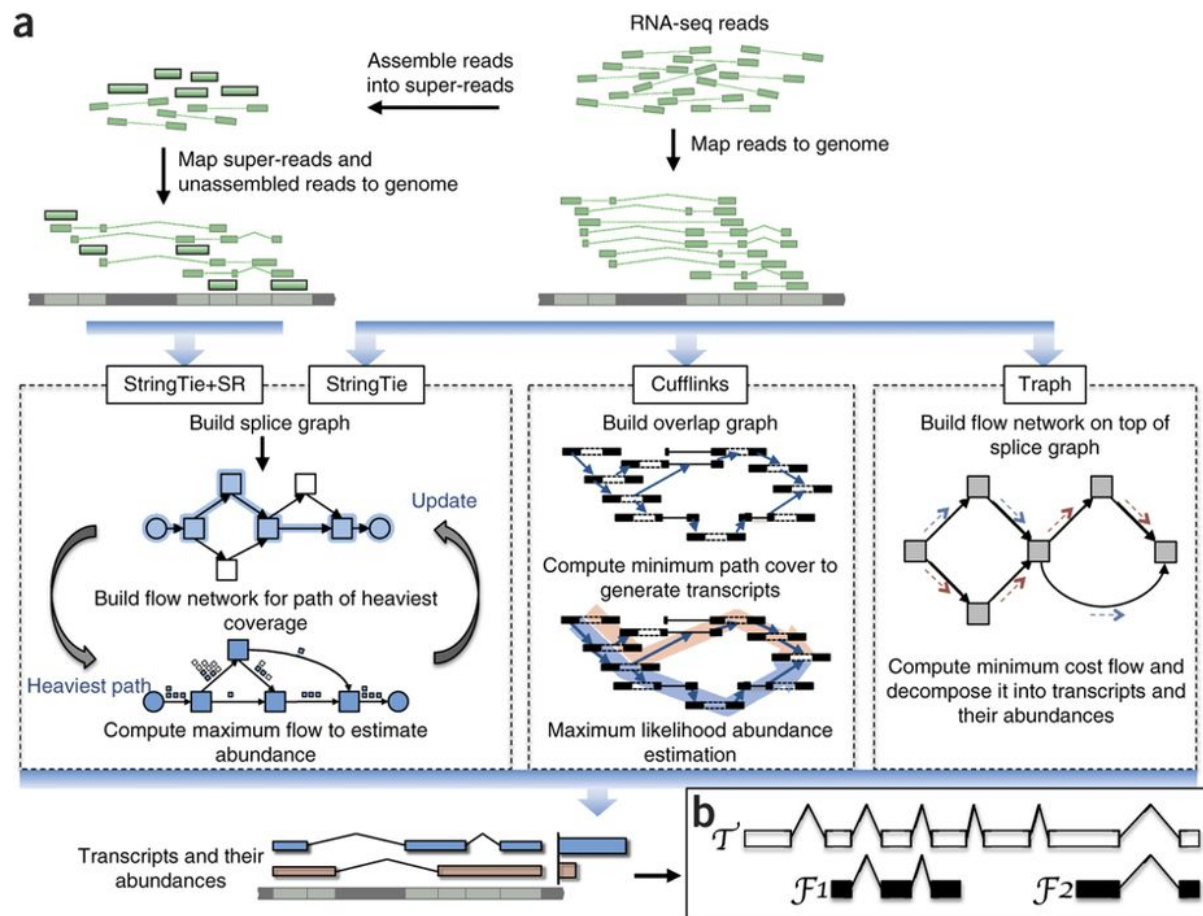
Output of `head -n 20 lmonocytogenes_genome.sam | less -S`

**Samtools** is a suite of programs for interacting with high-throughput sequencing data. With Samtools we are going to be able to perform some operations like reading, writing/editing, indexing, viewing SAM/BAM/CRAM format files.

In our Docker container, there are 2 different versions of Samtools. We have to use Samtools 1.9, and therefore, we have to export the right path to our environment with:

And converting our SAM file to BAM and sorting it. [Here](#) you can find a nice explanation of the objective of convert and sort your alignment data.

The main **output** of the program is a GTF file containing the structural definitions of the transcripts assembled by StringTie from the read alignment data.



Overview of the flow of the StringTie algorithm

To run *stringtie* and finish our analysis, please type in your Docker terminal:

```
stringtie /RESULTS/lmonocytogenes_genome_sorted.bam -G
/SAMPLES/lmonocytogenes_genome.gff -o /RESULTS/stringtie_all_output.gtf -A
/RESULTS/stringtie_cov_output.gtf
```

And finally, explore your results with:

```
ls -lh
```

```
root@b57cd699ff74:/RESULTS# ls -lh
total 1.7G
-rw-r--r-- 1 nobody 10000 5.0M Sep  2 10:24 lmonocytogenes_genome.1.bt2
-rw-r--r-- 1 nobody 10000 719K Sep  2 10:24 lmonocytogenes_genome.2.bt2
-rw-r--r-- 1 nobody 10000  17 Sep  2 10:24 lmonocytogenes_genome.3.bt2
-rw-r--r-- 1 nobody 10000 719K Sep  2 10:24 lmonocytogenes_genome.4.bt2
-rw-r--r-- 1 nobody 10000 374M Sep  2 10:27 lmonocytogenes_genome.bam
-rw-r--r-- 1 nobody 10000 5.0M Sep  2 10:24 lmonocytogenes_genome.rev.1.bt2
-rw-r--r-- 1 nobody 10000 719K Sep  2 10:24 lmonocytogenes_genome.rev.2.bt2
-rw-r--r-- 1 nobody 10000 477M Sep  2 10:26 lmonocytogenes_genome.sam
-rw-r--r-- 1 nobody 10000  66M Sep  2 10:29 lmonocytogenes_genome_sorted.bam
-rw-r--r-- 1 nobody 10000 238K Sep  2 10:23 lmonocytogenes_read1_fastqc.html
-rw-r--r-- 1 nobody 10000 269K Sep  2 10:23 lmonocytogenes_read1_fastqc.zip
-rw-r--r-- 1 nobody 10000 160M Sep  2 10:23 lmonocytogenes_read1_paired.fastq
-rw-r--r-- 1 nobody 10000 244K Sep  2 10:23 lmonocytogenes_read1_paired_fastqc.html
-rw-r--r-- 1 nobody 10000 283K Sep  2 10:23 lmonocytogenes_read1_paired_fastqc.zip
-rw-r--r-- 1 nobody 10000 5.5M Sep  2 10:23 lmonocytogenes_read1_unpaired.fastq
-rw-r--r-- 1 nobody 10000 241K Sep  2 10:23 lmonocytogenes_read2_fastqc.html
-rw-r--r-- 1 nobody 10000 272K Sep  2 10:23 lmonocytogenes_read2_fastqc.zip
-rw-r--r-- 1 nobody 10000 154M Sep  2 10:23 lmonocytogenes_read2_paired.fastq
-rw-r--r-- 1 nobody 10000 251K Sep  2 10:23 lmonocytogenes_read2_paired_fastqc.html
-rw-r--r-- 1 nobody 10000 297K Sep  2 10:23 lmonocytogenes_read2_paired_fastqc.zip
-rw-r--r-- 1 nobody 10000 436K Sep  2 10:23 lmonocytogenes_read2_unpaired.fastq
-rw-r--r-- 1 nobody 10000 955K Sep  2 10:30 stringtie_all_output.gtf
-rw-r--r-- 1 nobody 10000 379K Sep  2 10:30 stringtie_cov_output.gtf
```

Results of our hands-on