

# Computational biology:

# **Salmonella outbreak**

---

MARKOVIĆ Milena, VONDRAČEK Dušan  
October, 2020



## Table of contents

<b>Problem analysis</b>	<b>2</b>
Terminology and some introduction	2
Task description	3
Costs	3
<b>Solving the problems</b>	<b>5</b>
Removing sequencing errors	5
Estimating coverage and its distribution	5
Determining the right k	6
Finding SNPs	10
Protein mutation	13
Protein structure prediction	13

## Problem analysis

A violent bacterial outbreak is currently happening, killing tons of people, and the usual antibiotics have absolutely no effect. TATFAR makes a call for developing tools that answer the current crisis but that can be used for further events. The specifications are:

- One tool that takes two simple FASTA sequencing files (Illumina reads) and outputs a list of SNPs
- One tool that takes a multiple sequence alignment in standard FASTA file format and outputs a protein structure.
- Make a proof of concept of the tools used for the tasks above on the current AMR crisis (antimicrobial resistance) by understanding what is the difference between the strains, what gene(s) is involved, build a model of the protein structure associated to the gene and give a possible explanation.

Firstly we make an estimation of the costs (DNA sequencing and also our workload) justified by a basic description of the piece of software we plan to develop, and

## Terminology and some introduction


**FASTA:** In bioinformatics and biochemistry, the FASTA format is a text-based format for representing either nucleotide sequences or amino acid (protein) sequences, in which nucleotides or amino acids are represented using single-letter codes.

Our two FASTA files contain almost 2 million reads each.

**Reads:** In DNA sequencing, a read is an inferred sequence of base pairs (or base pair probabilities) corresponding to all or part of a single DNA fragment. As technical considerations make it impossible to sequence very long pieces of DNA all at once, instead, many overlapping small pieces (reads) are sequenced.

In our case, the typical read size is around 250 base pairs.

The problem arises when we need to assemble all these reads into one sequence. The locations of the fragments within the genome and with respect to each other are not generally known, so to figure out if a read contains an error or not, we need to take into account that the number of the base pairs in reads is much greater than the number of the base pairs in the whole genome. This means that reads overlap, which can lead to reading errors. This is one of the problems we will be talking about in this work.



**K-mers:** subsequences of length k contained within a biological sequence. Primarily used within the context of computational genomics and sequence analysis, in which k-mers are composed of nucleotides (in our case A, T, G, and C), k-mers are capitalized upon to assemble DNA sequences, improve heterologous gene expression, identify species in metagenomic samples, and create attenuated vaccines.

We have yet to decide the size of kmers that suits us best.

**SNP:** A **single-nucleotide polymorphism** is a substitution of a single nucleotide at a specific position in the genome, that is present in a sufficiently large fraction of the population (e.g. 1% or more).

**BLAST:** The **Basic Local Alignment Search Tool** is an algorithm and program which finds regions of local similarity between sequences. It compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. BLAST can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.

**Nucleotide 6-frame translation-protein(blastx):** This program compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.

## Task description

The problem can be broken down into several steps.

First off, Illumina is the main technology used nowadays for sequencing. It produces short reads of around 150-250 base pairs and the produced data is reliable with less than 1% of sequencing errors. Those errors can be considered as uniformly distributed, and containing only mutations, meaning substituting a letter by another one.


The starting point of the analysis are two FASTA files containing genetic sequences of two salmonella strains - one that is resistant to tetracycline, one that is not.

The first piece of software should compare these two strains and find all single-nucleotide polymorphisms in order to find mutations. To do so, the sequences first need to have the sequencing errors removed in each FASTA file. Only after that can we compare the two genomes in order to find SNP-s.

## Costs

According to the National Human Genome Research Institute, the cost of the first DNA sequencing (at the beginning of this century) was estimated at a whopping \$300 million!

Based on data collected by NHGRI from the Institute's funded genome-sequencing groups, the cost to



generate a high-quality 'draft' human genome sequence had dropped to ~\$14 million by 2006. Luckily for us (and everyone else), we are not in 2000 nor 2006 anymore! Nowadays, the cost to sequence a human genome is less than \$1000!

Since we are responsible citizens and trusted collaborators, before getting the call outcome, we have already sent two strains of the DNA for sequencing and are waiting for Illumina to finish their part of the work.

So, if our calculation is right, the cost of DNA sequencing will be around \$2000 (to simplify, we will say the cost is the same in euros - 2000€).

As for our salaries: Glasdoor.fr indicates that an average monthly salary for a Data Scientist Intern in Paris is 1414€. Since there are two of us in the team (instead of the usual 3-person team), we would demand a slightly higher than average salary, let's say 2000€ (as we value our effort and time invested). The length of the project is around a month, so the total cost for our workload would be around 4000€. A bonus after a job well done would be highly appreciated.

In the end, the first estimation of the costs is around 6000€ (plus the bonus). Quite cheap, eh?

## Solving the problems

### Removing sequencing errors

Before being able to compare the two FASTA files, we need to “clean” our files - throw out the sequencing errors.

We know that our Illumina reads contain around 1% of sequencing errors and that those errors can be considered as uniformly distributed. To figure out if a read contains an error or not, we need to understand that reads overlap, which can lead us to the reading errors. This is where we start talking about k-mers (subsequences of length k) and the sequencing **coverage**.

### Estimating coverage and its distribution

As defined on the Illumina website, coverage describes the average number of reads that align to, or “cover” known reference bases. In our case, coverage is the number of unique reads which contain a k-mer in the reconstructed sequence.

The Lander/Waterman equation is a method for computing coverage. The general equation is:

$$C = \frac{L * N}{G}$$

where  $C$  stands for coverage,  $G$  is the genome length and  $L$  is the read length, whereas  $N$  is the number of reads. And since we are working with DNA sequences (which contain 2 strands) the reads will contain both of the strands, and thus the coverage defined above is the true coverage multiplied by 2.

In other words, for our specific case, we have:

$$L = 250$$

$$G = 5.000.000$$

$$N = 2.000.000$$

Therefore it is easy to compute that the average coverage would be around  $C = 50$ .

Now, the number of times a base or a k-mer is sequenced follows the Binomial  $B(p, n)$  distribution with  $p = \frac{L}{G}$ , which is very small, and  $n = N$ , which is very big. Thus, the Binomial distribution can be approximated with the Poisson distribution with the parameter  $\lambda = n * p$ . In addition, we can easily notice that lambda is the same as coverage  $C$ , therefore we can conclude that the number of times a k-mer is sequenced follows the Poisson distribution with parameter  $C = 50$ .

### Determining the right $k$

Analyzing the DNA sequences can mainly be done by counting k-mers. The process consists of counting how many times a subsequence of length  $k$  is contained inside of the whole sequence. By doing this, we get many different subsequences and their counts. As the process is very expensive to compute, we should be careful that  $k$  is not too small. In addition, if  $k$  is not right, the function wouldn't look anything like Poisson distribution. So, to find the right  $k$ , we experimented with different values:

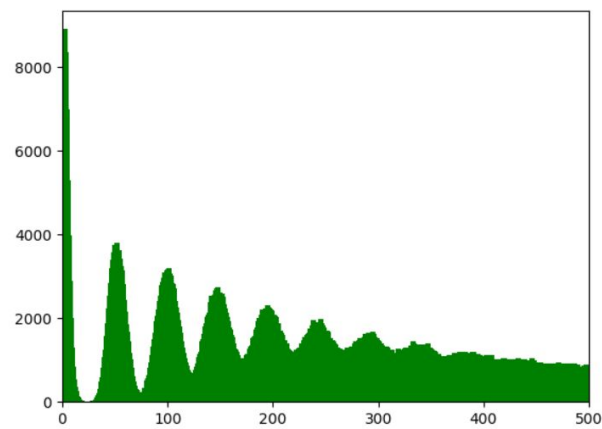


Figure 1. Distribution of k-mers for  $k=10$

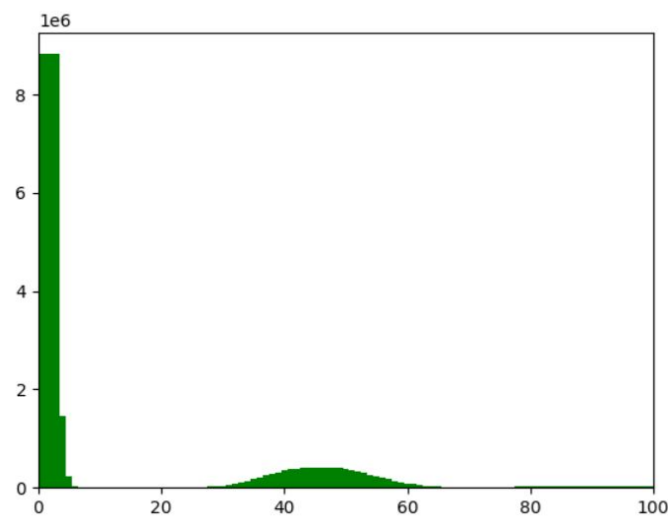


Figure 2. Distribution of k-mers for  $k=13$

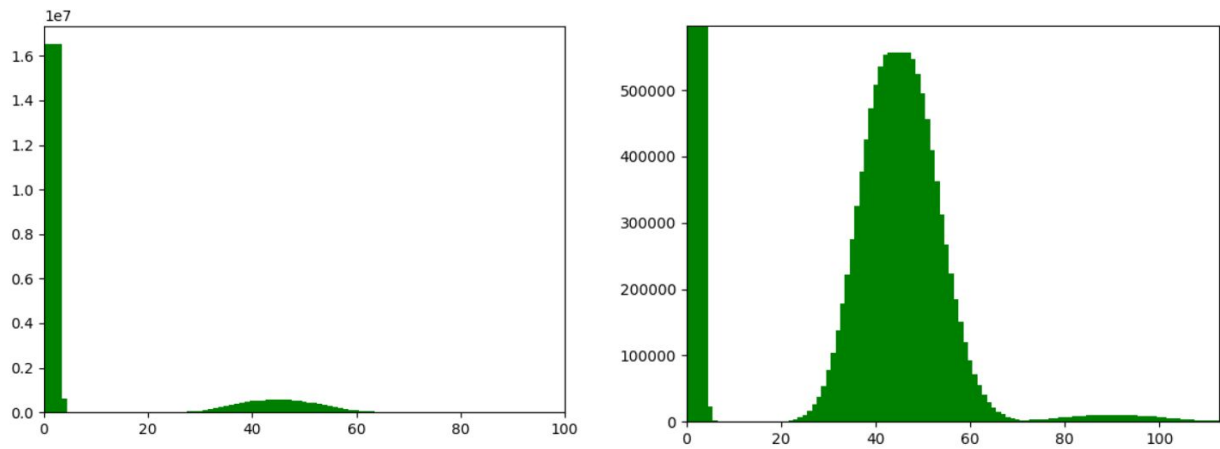


Figure 3. Distribution of k-mers for  $k=15$  (right: whole chart, left: zoomed in on the non-errors)

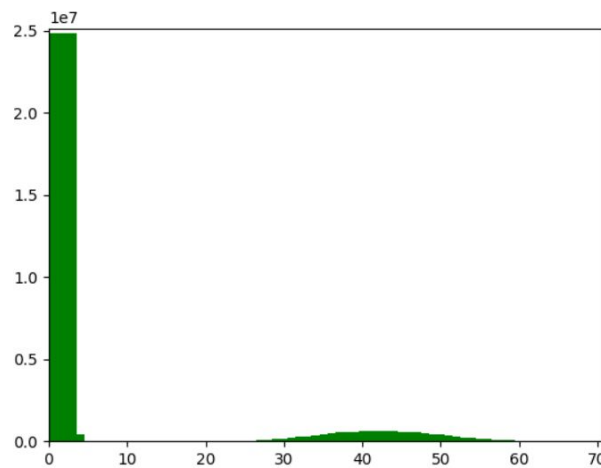


Figure 4. Distribution of k-mers for  $k=24K=17$  the ratio errors:non-errors starts to change in favor of

As we can conclude from the figures above, the right value for the length of k-mers is 15. If the value is too small, like in figures 1 and 2 (notice that there is another Poisson “hill” in (2)), we have multiple Poisson distributions, which is an indicator to raise the length of  $k$ ; and for a value larger than 15, the number of unique k-mers rises which is also not good.

So, how do we locate the sequencing errors? The answer is quite simple: errors usually change high-count k-mers into low-count k-mers. As pictured below, when an error occurs it will turn frequent k-mers to infrequent k-mers, since the error will happen in one of many overlapping reads.





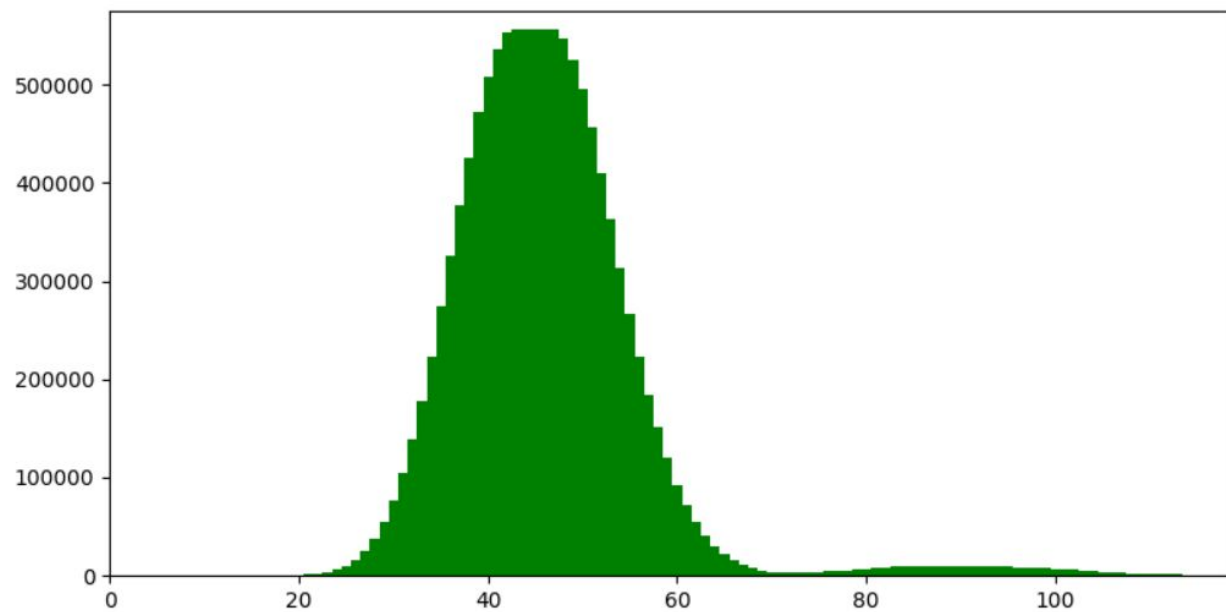
Sequence without errors:		Sequence with one error:	
Read (20bp) :	CGCGAACACTGGTCACATGT	Read (20bp) :	CGCGAACAT <del>T</del> GGTCACATGT
	CGCGAACAC ————— 9		CGCGAACAT ————— 1
	GCGAACACT ————— 11		GCGAACATT ————— 2
	CGAACACTG ————— 11		CGAACAT TG ————— 2
	GAACACTGG ————— 9		GAACAT TGG ————— 3
	AACACTGGT ————— 8		AACAT TGGT ————— 1
	ACACTGGTC ————— 10		ACAT TGGTC ————— 2
	CACTGGTCA ————— 11		CAT TGGTCA ————— 1
	ACTGGTCAC ————— 12		AT TGGTCAC ————— 2
	CTGGTCACA ————— 9		<del>T</del> TGGTCACA ————— 2
	TGGTCACAT ————— 9		TGGTCACAT ————— 9
	GGTCACATG ————— 11		GGTCACATG ————— 11
	GTCACATGT — 10		GTCACATGT — 10

The sequence from the picture shows the first 20 base pairs from the hands-on reference data (with made up numbers of repetitions - just as an explanation). The left side indicates that the counts of all the 9-mers are similar (around some average score) whereas the right side contains a part where there is obviously an error since the count of the 9-mers is much below average.

If we now turn back to our problem, we can assume that most of the k-mers which are more unique (located more to the left on the histogram) are there because of sequencing errors, so we will focus only on the ones without an error. We assumed counts 1-4 contain errors for sure, in accordance with the professor.

Then, we computed the coverage by observing the highest repeated k-mer group, and found it to be 47. So finally, the sequencing error removal consists of “removing” k-mers who had a high probability to be a reading error with respect to the Poisson distribution with parameter  $C = 47$ . In other words, we deemed the k-mers which have a probability lower than the error rate  $\eta = 1\%$  to contain reading errors. We will call SNPs only if the k-mers have a higher probability than the error rate.

After we focus only on the k-mers which are deemed not to contain sequencing errors, we get a solid Poisson distribution (shown on figure 5) and can move on to the next part of our work.



*Figure 5. Distribution of k-mers after removing k-mers with occurrence count  $\leq 8$*

Now that we have files without errors we can move on to the final step.

## Finding SNPs

In the previous step, we removed k-mers that most probably contain sequencing errors. This produced two files: for each strain of the bacteria that we are observing, one file that contains all of their k-mers that are left (after removing the errors) and their occurrence counts. To find the mutation that we are looking for, we now need to compare these two k-mer lists.

What we want to do here is to check what k-mers that appear in one strain and not in the other (and vice versa). To do this, we will load both k-mer lists into python dictionaries, and iterate through each of them. For every k-mer from one strain, we will check whether it exists in the dictionary of the other strain. If not, we will save it to a list of k-mers that potentially contain SNP-s.

After performing this on our files, we got the following lists of k-mers unique to each strain:

<i>Salmonella Enterica</i>		<i>Salmonella Enterica Variant</i>	
CTTCTGGGCGAGTTT	50	AGCTTCTGGGCGAGG	47
TTCTGGGCGAGTTTA	49	GCTTCTGGGCGAGGG	47
TCTGGGCGAGTTTAC	49	CTTCTGGGCGAGGGG	47
CTGGGCGAGTTTACG	49	TTCTGGGCGAGGGGA	47
TGGGCGAGTTTACGG	48	TCTGGGCGAGGGGAC	47
GGGCGAGTTTACGGG	48	CTGGGCGAGGGGACG	46
GGCGAGTTTACGGGT	48	TGGGCGAGGGGACGG	47
GCGAGTTTACGGGTT	48	GGGCGAGGGGACGGG	48
CGAGTTTACGGGTTG	48	GGCGAGGGGACGGGT	48
GAGTTTACGGGTTGT	49	GCGAGGGGACGGGTT	48
AGTTTACGGGTTGTT	48	CGAGGGGACGGGTTG	48
GTTTACGGGTTGTTA	48	GAGGGGACGGGTTGT	48
TTTACGGGTTGTTAA	49	AGGGGACGGGTTGTT	48
TTACGGGTTGTTAAA	49	GGGGACGGGTTGTTA	48
TACGGGTTGTTAAAC	49	GGGACGGGTTGTTAA	48
		GGACGGGTTGTTAAA	47
		GACGGGTTGTTAAAC	47
GTTTAACAACCCGTA	45	GTTTAACAACCCGTC	34
TTTAACAACCCGTAA	46	TTTAACAACCCGTCC	35
TTAACAACCCGTAAA	46	TTAACAACCCGTCCC	36
TAACAACCCGTAAAC	46	TAACAACCCGTCCCC	36
AACAACCCGTAAACT	48	AACAACCCGTCCCCT	35
ACAACCCGTAAACTC	48	ACAACCCGTCCCCTC	34
CAACCCGTAAACTCG	48	CAACCCGTCCCCTCG	34
AACCCGTAAACTCGC	48	AACCCGTCCCCTCGC	33
ACCCGTAAACTCGCC	48	ACCCGTCCCCTCGCC	33
CCCGTAAACTCGCCC	49	CCCGTCCCCTCGCCC	32

CCGTAAACTCGCCCA	48	CCGTCCCCTCGCCCA	32
CGTAAACTCGCCCAG	48	CGTCCCCTCGCCCAG	32
GTAAACTCGCCCAGA	47	GTCCCCTCGCCCAGA	32
TAAACTCGCCCAGAA	46	TCCCCTCGCCCAGAA	32
AAACTCGCCCAGAAG	45	CCCCTCGCCCAGAAG	31
		CCCTCGCCCAGAAGC	31
		CCTCGCCCAGAAGCT	32
GGTTCGTGCGTCACC	15	AAAGGCTGATACATT	10
GTTTCGTGCGTCACCC	14	AAGGCTGATACATTA	10
TTCGTGCGTCACCCT	13	AGGCTGATACATTAA	11
TCGTGCGTCACCCTT	13		
CGTGCGTCACCCTTC	13		
GTGCGTCACCCTTCA	13		
TGCGTCACCCTTCAT	13		
GCGTCACCCTTCATG	13		
CGTCACCCTTCATGC	12		
GTCACCCTTCATGCA	12		
TCACCCTTCATGCAG	11		
CACCCTTCATGCAGG	11		
ACCCTTCATGCAGGG	11		
CGACGTGCGCCAACA	9	TGTGGCTGGTAACTC	9
GACGTGCGCCAACAA	9	CAAAGGCTGATACAT	9
ACGTGCGCCAACAAT	9	CGCCAGCGGGGATAT	9
CGTGCGCCAACAATC	9		
GTGCGCCAACAATCA	9		
TGCGCCAACAATCAT	9		
GCGCCAACAATCATT	9		

We can observe that in both lists, some of the k-mers can be split into “batches” by grouping them so that they have k-1 BP of overlap - with the exception of the k-mers in the last row of the variant strain, that do not have any other k-mers k-1 BP overlap. We will not consider these k-mers further. By concatenating these batches, we further have the following sequences:

	<i>Salmonella Enterica</i>		<i>Salmonella Enterica Variant</i>
1A	CTTCTGGGCGAGTTTACGGGTTGTTAAAC	1B	AGCTTCTGGGCGAGGGACGGGTTGTTAAAC
2A	GTTTAACAACCCGTAAACTCGCCCAGAAG	2B	GTTTAACAACCCGTCCCCTCGCCCAGAAGCT
3A	GGTTCGTGCGTCACCCTTCATGCAGGG	3B	AAAGGCTGATACATTAA
4A	CGACGTGCGCCAACAATCATT		

Right at first glance, we can observe that the sequences in the first two rows are partial matches: between sequences 1A and 1B, pairs highlighter in red are a match and in yellow they are different. Same goes for 2A and 2B - green pairs are a match, while yellow pairs are different.

Upon further investigation, we can also see that 1A and 2A are reverse complements, and as are 1B and 2B:

	2A	2B
Original	GTTTAAACAACCCGTAAACTCGCCCAGAAG	GTTTAAACAACCCGTCCCCTCGCCCAGAAGCT
Reverse	GAAGACCCGCTCAAAATGCCCAACAATTTG	TCGAAGACCCGCTCCCCTGCCCAACAATTTG
Complement	CTTCTGGGCGAGTTTACGGGTTGTTAAAC	AGCTTCTGGGCGAGGGGACGGGTTGTTAAAC

This serves as strong evidence that this is the sequence that contains the SNP mutation that we are looking for.

As for the other 3 detected unique sequences, due to the low occurrence count of the k-mers, that is very close to the cutoff threshold for error detection, as well as the fact that even when we find their complements and reverse them, they do not match any found sequence from the opposite list, we will not consider them either.

Our final conclusion is that the sequence in the salmonella variant strain that we were looking for is **AGCTTCTGGGCGAGGGGACGGGTTGTTAAAC**. We can further search through the original FASTA file and find reads that contain this sequence in order to acquire a larger region that contains more of the mutated gene for further analysis.

## Protein mutation

**AGCTTCTGGGCGAGGGGACGGGTTGTTAAAC**

This sequence should be the wanted SNP. We used this sequence to search for reads that contain it in the original salmonella variant FASTA file and extract them to a separate file. Then, we used this file and used it to perform a search on **blastx**. The reason why we took the entire reads and not just the sequence is that blastx can't work with sequences that are too small (it needs a sequence containing more than ~100 nucleotides). Some of the reads might contain reading errors somewhere else than the sequence, but since there will be multiple reads of the same part of the genome, the search engine will be able to identify the corresponding gene anyway.

It didn't take long for blastx to find sequences producing significant alignments, and all of them point to the same organism - *Salmonella enterica* (TetR family transcriptional regulator). "TetR is the repressor of the tetracycline resistance element; its N-terminal region forms a helix-turn-helix structure and binds DNA. Binding of tetracycline to TetR reduces the repressor affinity for the tetracycline resistance gene (tetA) promoter operator sites." [5]



## Literature:

[1] Illumina website [\[link\]](#)

[2] Estimating sequencing coverage. Illumina [\[link\]](#)

[3] Poisson Distribution in Genome Assembly. University of Illinois courses [\[link\]](#)

[4] Understanding Sequencing Reads: Introduction. Mark Dunning [\[link\]](#)

[5] Tetracycline repressor protein. UniProt [\[link\]](#)

[6] various articles from Wikipedia