University Grenoble-Alpes

Salmonella Outbreak

Students
Kseniia Varlamova
Ignat Sabaev

Teacher
Clovis Galiez

October 18, 2024

1 Problem Description

The project solves the problem of searching for SNPs in the genome. Two sequences are input: the original genome and the genome with mutations. The data are presented in fasta format. It is necessary to find SNPs in the mutated genome. It is known that 1% of uniformly distributed errors are allowed in the data. In particular, the project proposes to solve the problem for Salmonella bacteria. Full information about the task is available in the gitlab `https://gitlab.ensimag.fr/galiezc/protein-structure-prediction`.

2 Statistical Model

To find data with errors, it is suggested to break the data into k-mers and count their occurrence. Error k-mers will occur much less frequently than error-free k-mers. For this purpose, let us consider the distribution of k-mers.

What we have in advance:

- $L = 250$ — length of one read in data (number of nucliotides);
- $\eta = 0.01$ — error probability;
- $N \approx 2 \cdot 10^6$ — number of reads in data;
- $G = 2 \cdot 4857450$ — true genome length due to `https://www.ncbi.nlm.nih.gov/datasets/genome`. In practice, we use multiplication by 2 because we have both straight and reversed reads in our data.

Let C_k — number of reads in data, overlapping particular k-mer. Then C_k has a distribution:

$$C_k \sim Bin\left(N, \frac{L-k+1}{G-L+1}\right).$$

Let Y_k — number of times we should find particular k-mer. Then Y_k has a distribution:

$$Y_k \sim Bin\left(C_k, (1-\eta)^k\right).$$

We will estimate this distribution with mathematical expectation of C_k:

$$\hat{Y_k} \sim Bin\left(E[C_k], (1-\eta)^k\right) = Bin\left(N \cdot \frac{L-k+1}{G-L+1}, (1-\eta)^k\right).$$

3 Min and Max thresholds choosing

To separate k-mers into 'error' and 'solid' we construct confidence intervals for the maximum and minimum number of k-mers. To do this, we use the parametric bootstrap method. We sample $N_{smpl} = 10000$ times from the theoretical distribution presented above. The size of each sample was taken corresponding to the number of unique graphically valid k-mers. In this way, we collected samples for the maxima and minima. Then

construct confidence intervals for them with a significance level of $\beta = 0.95$ using the formulas:

$$B_{left} = \frac{1-\beta}{2} \cdot N_{smpl},$$

$$B_{right} = \frac{1+\beta}{2} \cdot N_{smpl},$$

where B_{left}, B_{right} are left and right boundaries of the confidence interval, respectively. For the minimum we need left boundary B^{min}_{left}, for the maximum — right boundary B^{max}_{right}. So we're gonna assume

- solid such k-mers whose number in the data is located between B^{min}_{left} for minima and B^{max}_{right} for maxima;
- error such k-mers whose number is less than B^{min}_{left};
- outliers such k-mers whose number exceeds B^{max}_{right} for maxima. Such outliers may be justified by the fact that some k-mers may be far from unique, being part of a large number of genes. Their number is especially large when k is small.

In practice, a larger value was taken for Max threshold, due to the small bias in the empirical distribution. This is shown in more detail in jupyter notebook *main.ipynb*.

4 Empirical k choosing

A range from 11 to 25 was chosen for the initial estimation of k. Smaller values of k provoke a large number of overly false-non-unique k-mers. Larger values of k, on the contrary, generate unique k-mers that become impossible to distinguish from erroneous ones. For all values of k in this interval (in steps of 2), k-mers were counted. The empirical distribution was then examined in conjunction with the theoretical distribution. In particular, we considered the number of k-mers whose number does not exceed the bounds that initially estimate the left and right limits.

Thus, for k=11 and k=17 we have results shown in Fig. 1. The distribution starting with 17 is almost constant. At a value of 15, there is still a small number of data logically to the right of the theoretical values. However, as k increases, bias accumulates. Taking into account all factors, we take $k = 17$. The empirical data still has larger variance, but the distribution form at $k = 17$ became closer to the theoretical one. More information and visualisation are shown in *main.ipynb*.

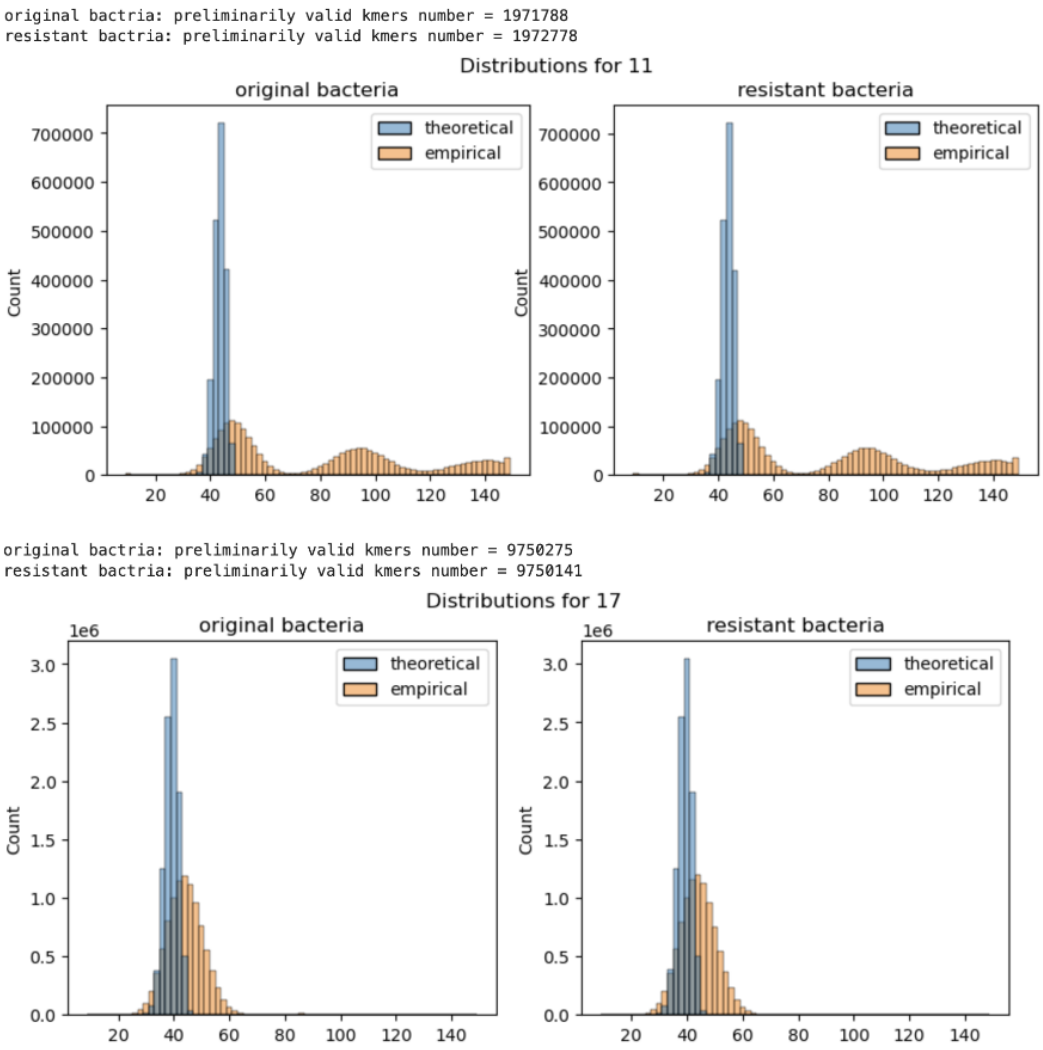

Figure 1: 11-mers and 17-mers distributions

5 SNP search algorithm

In order to find a mutation with already selected thresholds and k, an algorithm was developed:

1. Data preprocessing.
   (a) Read data, translate into standard python types.
   (b) Removal of repetitive reads in both original and mutant bacteria (repetitions can be justified by the way the data was collected).
   (c) Partitioning the data into kmers, creating a dictionary {kmer: number in the data}.
   (d) Selecting 'solid' kmers, according to chosen thresholds.
2. Candidates search.
   (a) Collection of kmers that are selected for one bacterium and not selected for another.
   (b) Combine such kmers into sequences, since one mutation generates k such potential kmers.
3. Original-mutated pairs correlation. Correlation of sequences from the original and mutated bacterium was developed using *CountVectorizer*. The distance between the sequences of one and another bacterium was calculated as the number of different characters at the same positions (for different sequence lengths, the position generating the minimum distance was searched). The top-2 candidates from the mutated bacterium were searched for each sequence from the original bacterium, the distance between which does not exceed the expected number of mutations (in this case we set this number as $[k/3]$).
4. Postprocessing. Processing output in readable format. Search for reads in the original data to lengthen the final sequences so that it is possible to search for genes across databases.

6 Time costs

Most of the time was spent on finding the optimal hyperparameters (k, min and max thresholds). Each variant k required $O\left(N \cdot (L - k + 1)\right)$ time linear of the input length. Eight variants of k were tried, this took about 100 minutes. It took 15 minutes to find the threshholds. The inference step takes 3-7 minutes and also depends on the length of the input.

7 Algorithm validation

In order to verify the algorithm viability, a small test was performed. Another mutation was synthesised. We took the same genome for the original bacterium and selected a random valid kmer from it. For the test 'mutant' bacterium, we changed two consecutive nucleotides for all these kmer to two other nucleotides. We then passed this data to the algorithm. As expected, the algorithm found sequences containing these kmers, indicating that the algorithm worked.

8 Biological interpretation

We found two nucleotide sequences:

1. AGGTTTAACAACCCGT**CCC**CTCGCCCAGAAGCTA (in mutated becteria)
   AGGTTTAACAACCCGT**AAA**CTCGCCCAGAAGCTA (in original becteria)

2. CTAGCTTCTGGGCGAG**GGG**ACGGGTTGTTAAACC (in mutated becteria)
   CTAGCTTCTGGGCGAG**TTT**ACGGGTTGTTAAACC (in original becteria)

When gueries on `https://blast.ncbi.nlm.nih.gov/Blast.cgi` we get matches with the accuracy up to the found three nucleotides. In particular:

- Salmonella enterica subsp. enterica serovar Kentucky strain CVM N38232 plasmid pN38232-1;
- Salmonella enterica subsp. enterica serovar Typhimurium strain SJTUF10057;
- Salmonella enterica subsp. enterica serovar Typhimurium strain SJTUF10484.

This further confirms that this is where the mutation is.

After a few queries of the database, we can guess that the gene is named *mig-14*: `https://www.ncbi.nlm.nih.gov/protein/2518927816`; `https://pmc.ncbi.nlm.nih.gov/articles/PMC135090`.