Department of Electronic & Telecommunication Engineering
Faculty of Engineering
University of Moratuwa

Genomic Signal Processing

Promoter Discovery in Bacteria Azotobacter chroococcum

Bandara D.M.D.V.
Undergraduate (Biomedical engineering)
Department Electronic and Telecommunications
Faculty of Engineering
University of Moratuwa

December 19, 2024

Table of Content

List of Figures

1 Introduction and Background

Azotobacter chroococcum is a nitrogen-fixing bacterium found in soil, it is can convert atmospheric nitrogen (N_2) into ammonia through nitrogen fixation. This promotes soil fertility, making *A. chroococcum* significant in agriculture and environmental microbiology. Understanding the genetic regulation of this bacterium gives an idea about nitrogen fixation mechanisms and overall metabolic processes.

Promoters are DNA sequences located near the start of genes, responsible for initiating transcription. Transcription allows the bacterium to perform different functions by producing different proteins. Promoters initiate binding sites for RNA polymerase and other regulatory proteins and control gene expression. Promoters are often found upstream of coding regions, and their identification is essential for understanding gene regulation and expression patterns in bacteria.

In bacterial genomes, common promoters have conserved sequences, such as the $TATA$ box and the $TTGACA$ sequence. Another type of promoter, which is the focus of this study, is the "$WWWW$" promoter, where "W" represents either adenine (A) or thymine (T). Promoter identification methods typically involve sequence analysis and computational techniques. Two prominent approaches are local alignment search and statistical alignment:

- **Local Alignment Search:** This method identifies regions of similarity between a given query sequence (such as the "$WWWW$" motif) and target sequences. This finds potential promoters by comparing sequences and scoring local matches.
- **Statistical Alignment:** This technique involves calculating a position probability matrix (PPM) based on the frequencies of nucleotides at specific positions within a set of sequences. The PPM can then be used to identify statistically significant patterns, providing a probabilistic approach to finding promoters.

This project compares these two methods for identifying "$WWWW$" promoters in the *Azotobacter chroococcum* genome. Specifically, the study aims to evaluate the effectiveness of local alignment search and statistical alignment in detecting promoters, analyzing the percentage of detected promoters, lengths and the distribution of their positions.

2 Extracting the Genes

For the assigned GenBank accessions (CP011835.1) of nitrogen-fixing bacteria of the genus *Azotobacter chroococcum*, the gene assembly data were downloaded [1]. Of these, the .fna and .gtf files were used to obtain the genome and loci of genes, respectively.
The Biopython library SeqIO [2] was used to read the FASTA (.fna) file. For each sequence in the FASTA file, the ID, description, and length were printed to get an initial understanding of the data.

```
ID: NZ_CP011835.1
Description: NZ_CP011835.1 Azotobacter chroococcum strain B3 chromosome, complete genome
Sequence length: 4575910
--------------------------------------------------
ID: NZ_CP011837.1
Description: NZ_CP011837.1 Azotobacter chroococcum strain B3 plasmid pacX50dB3a
Sequence length: 74783
--------------------------------------------------
ID: NZ_CP011836.1
Description: NZ_CP011836.1 Azotobacter chroococcum strain B3 plasmid pacX50fB3
Sequence length: 306103
--------------------------------------------------
ID: NZ_CP011838.1
Description: NZ_CP011838.1 Azotobacter chroococcum strain B3 plasmid pacx50dB3b
Sequence length: 66259
--------------------------------------------------
```

Figure 1: FASTA (.fna) file sequence details

The loci of the genome, provided in the .gtf file, were read as a CSV file and printed to examine the details of the gene features and their positions.

	sequence_name	source	feature	start	end	score	strand	\
0	NZ_CP011835.1	RefSeq	gene	1467	2570	.	+	
1	NZ_CP011835.1	Protein Homology	CDS	1467	2567	.	+	
2	NZ_CP011835.1	Protein Homology	start_codon	1467	1469	.	+	
3	NZ_CP011835.1	Protein Homology	stop_codon	2568	2570	.	+	
4	NZ_CP011835.1	RefSeq	gene	2589	3692	.	+	

	frame	attribute
0	.	gene_id "ACG10_RS00010"; transcript_id ""; db_...
1	0	gene_id "ACG10_RS00010"; transcript_id "unassi...
2	0	gene_id "ACG10_RS00010"; transcript_id "unassi...
3	0	gene_id "ACG10_RS00010"; transcript_id "unassi...
4	.	gene_id "ACG10_RS00015"; transcript_id ""; db_...

Figure 2: .gtf file headers and details

The sense strand was extracted by selecting loci with "+" (implies sense direction) as the strand and "gene" as the feature.

	sequence_name	source	feature	start	end	score	strand	frame	\
0	NZ_CP011835.1	RefSeq	gene	1467	2570	.	+	.	
4	NZ_CP011835.1	RefSeq	gene	2589	3692	.	+	.	
8	NZ_CP011835.1	RefSeq	gene	3698	6118	.	+	.	
12	NZ_CP011835.1	RefSeq	gene	6425	7786	.	+	.	
16	NZ_CP011835.1	RefSeq	gene	8291	9853	.	+	.	

	attribute
0	gene_id "ACG10_RS00010"; transcript_id ""; db_...
4	gene_id "ACG10_RS00015"; transcript_id ""; db_...
8	gene_id "ACG10_RS00020"; transcript_id ""; db_...
12	gene_id "ACG10_RS00025"; transcript_id ""; db_...
16	gene_id "ACG10_RS00030"; transcript_id ""; db_...

Figure 3: Sense strand loci extracted from the .gtf file

From the "start" and "end" values obtained in Figure 3 above, the genome was partitioned into gene sequences. Since the goal is to find the promoter for each gene, 100 base pairs were selected before the start position and 3 base pairs after the end position. When the code was executed, 2457 sequences were extracted. To verify the correctness of the code, for each sequence, positions 100,101 and 102 were checked to see if they contained the A, T and G (methionine) respectively as the start codon. Since the start of the gene (coding region) is typically methionine, if the majority of sequences contained ATG, we could confirm that the code was functioning correctly. Code can be found in [3]

```
Methionine count: 1911
                                Not Methionine count: 546
percentage of methionine: 77.77777777777779%
```

Figure 4: Count of Methionine as the start codon for the extracted 2457 sequences

3 Implementation - Answering given five questions

3.1 Perform a standard local search (for an intact query) to locate the $WWWW$ promoter within each sequence. Obtain the percentage of genes with potential promoters and the distribution of the upstream position.

For the local search algorithm to identify the "$WWWW$" promoter in each sequence of the genome (2457 sequences), the first 100 base pairs upstream of the gene were selected. Temporarily, all "T" and "A" bases in the sequence were replaced with "W" bases to align the sequences for the promoter search. The last 12 base pairs were excluded from the search to leave sufficient codons (assumed to be 4) for transcription.
As we were tasked with performing an intact query match, the match score was set to 1, and the gap penalty was set to -2. The search for the "$WWWW$" promoter was then carried out in the reverse direction to identify the promoter closest to the gene.

I implemented three functions before defining the local back-propagation algorithm:

1. A function to convert all occurrences of "A" and "T" to "W" in a given sequence.
2. A function to check for a match or mismatch between two base pairs.
3. A function to iterate through the sequence and generate the score matrix.

Next, a trace-back function was defined to perform the local search, starting from the highest score (=4) closest to the gene. If "$WWWW$" was not found consecutively in the sequence, it was concluded that the sequence does not have the promoter. When the promoter was found, a list was created including,

- The sequence number,
- The aligned promoter region in the sequence, and
- The position of the promoter.

```
[[3, 'TTAA', 52], [4, 'AAAA', 13], [7, 'TTTA', 32], [9, 'TTTT', 72], [10, 'ATTA', 68],
percentage of promoters: 60.48026048026048
```

Figure 5: The created list and percentage of promoters found with local alignment

The created list containing 1486 sequences where the "$WWWW$" promoter was found, is used in subsequent sections of the project. Note that the percentage of promoters found directly corresponds to the percentage of sequences that will undergo transcription. After finding the promoter positions, a histogram was plotted to visualize the positions of the promoter distribution, across the 100 base pair locations. The distribution was obtained as shown.

```
1486 promoters found out of 2457
```

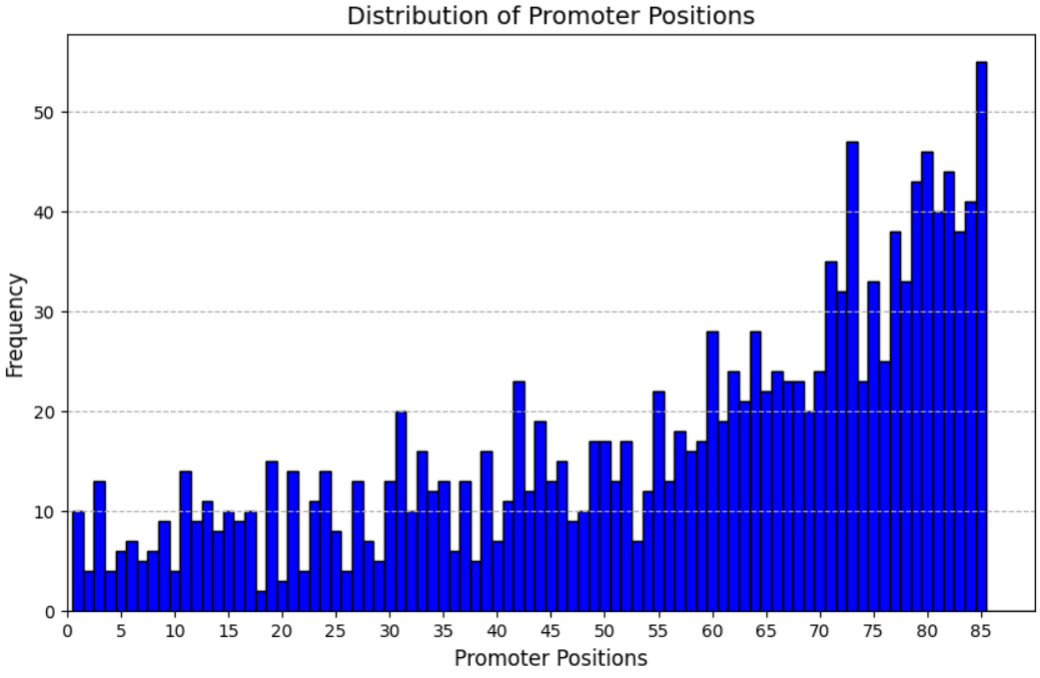

Figure 6: Position distribution of promoters through local alignment search

3.2 If a "$WWWW$" promoter is found find the number of consecutive "W"s. Obtain the distribution of the number of consecutive "W"s.

From the position list created earlier, a function was written to go through the sequences where the "$WWWW$" promoter was found. Around the identified promoters, the number of consecutive "W"s associated was calculated. A histogram was then plotted.

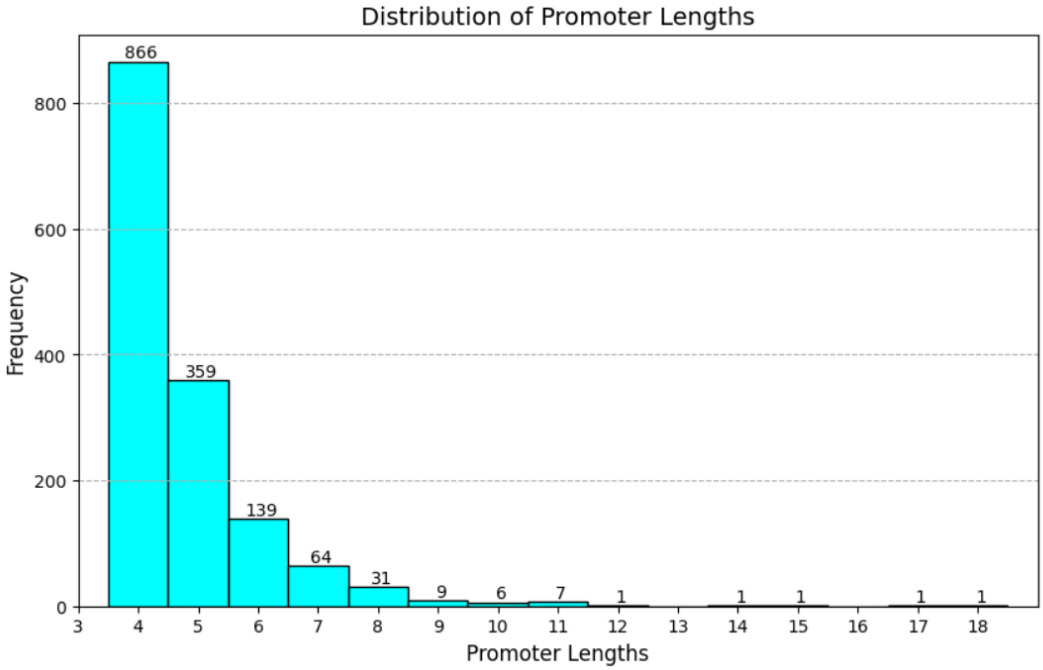

Figure 7: Promoters full length of consecutive "A"s and "T"s

3.3 Obtain the sequence starting from each upstream position of (section 3.1) which is 6 bases long (regardless of its C/G content). Calculate the position probability matrix.

From the positions obtained in section 3.1, I take 6 upstream sequences. This was repeatedly done for 1486 sequences where "$WWWW$" was detected. Then, the position probability matrix (PPM) was calculated for each of the 6 positions and 4 base pairs.
To calculate the PPM, one function was created to obtain the 6-base-pair-long upstream sequence, and another function was created to calculate the PPM and consensus score of the found sequence array.

```
      (1)    (2)    (3)    (4)    (5)    (6)
(A)  0.569  0.498  0.468  0.405  0.008  0.219
(C)  0.001  0.001  0.001  0.001  0.600  0.350
(G)  0.001  0.001  0.001  0.001  0.388  0.270
(T)  0.429  0.500  0.530  0.593  0.004  0.162

Consensus score: -3.976640016300933
```

Figure 8: PPM matrix for promoter length of 6

From the found PPM, the consensus score was calculated to be -3.976640016300933. This score is used as a baseline to calculate the normalized score for each alignment done below.

3.4 Using the position probability matrix of (section 3.3) obtain a statistical alignment of the same sequences.

Now, for the whole genome (2457 sequences), the first 100 base pairs were selected, ignoring the last 9 base pairs (this was done to match the positional distribution range by including 3 more base pairs in the promoter search). For these sequences, a window of length 6 was extracted, and the alignment score was calculated using the PPM. The normalized score (Norm. Score = alignment score - consensus score) is used as a metric to find the best-matching promoter. A threshold (-3 for my convenience) was set, and if the Norm. Score is greater than the threshold, the window is considered a possible promoter.
For this process, two functions were created: one to calculate the Norm. Score and another to find the statistical alignments.

```
[[3, 'AATTGA', 5], [4, 'AAAAGT', 13], [7, 'ATTTCA', 32], [9, 'TTTTCA', 72], [10, 'ATTACG', 68],
percentage of promoters: 60.48026048026048 for threshold value -6
```

Figure 9: The created list and percentage of promoters found with statistical alignment

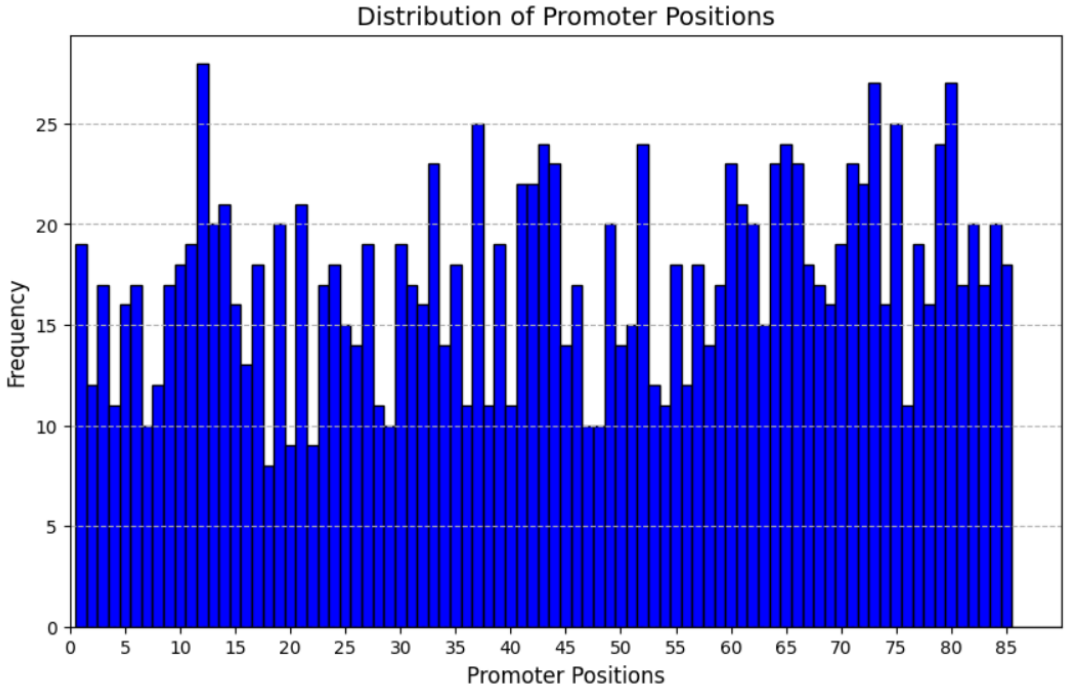

Figure 10: Position distribution of promoters through statistical alignment search

3.5 Compare the percentage of promoters detected and position distribution results of the two searches. Comment on the results.

Both local alignment and statistical alignment perfectly capture the presence of the "*WWWW*" promoter. Hence, the percentage of promoters located through both methods remains the same (60.48%).
By changing the threshold below -6, the program can detect over 95% of the promoters. However, this would include sequences without 4 consecutive "W"s. Additionally, changing the threshold between -3 and -6 does not result in a percentage change. This indicates that there is a clear gap in Norm. Score values between possible promoters and other 6-base-pair permutations.
From the distributions, it is clear that the promoter locations are biased in the local alignment search toward the condition used to select the alignments. Here, I used the condition to select the possible promoter closest to the gene. One can change this condition to select the one furthest from the gene, the one with the most consecutive "*W*"s, etc.
In contrast, statistical alignment identifies the promoter that maximizes the positional probability values. This means it finds the promoter with the highest likelihood of being a promoter by searching through the set of prior codons. Hence, statistical gene alignment provides a more accurate positional distribution for promoters.

4 Conclusion

In this study, the identification of "*WWWW*" promoters in the genome of *Azotobacter chroococcum* was explored using two computational approaches: local alignment and statistical alignment. Both methods successfully detected promoters with comparable accuracy in terms of identifying promoter presence. However, statistical alignment provided a more understanding of promoter positioning by maximizing positional probabilities, demonstrating its potential for more accurate promoter characterization in genomic studies.

References

[1] National Center for Biotechnology Information, "Genome Assembly: Azotobacter chroococcum NCIMB 8003." [Online]. Available: https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_002220155.1/

[2] Cock, P. J. A. and Antao, T. and Chang, J. T. and Chapman, B. A. and Cox, C. J. and Dalke, A. and Friedberg, I. and Hamelryck, T. and Kauff, F. and Wilczynski, B. and de Hoon, M. J. L., "Biopython: SeqIO Module." [Online]. Available: https://biopython.org/wiki/SeqIO

[3] D. Vinod, "Promoter search in bacteria," 2024, accessed: 2024-12-19. [Online]. Available: https://github.com/D-Vinod/Promoter_search_bacteria