Workshop Report No. 1: Motif Detection in Genetic Sequences

System Analysis

Teacher : Carlos Andres Sierra Virguez

Name: Brayan Camilo Gomez Rodriguez

Code: 20231020167

Systems Engineering

Universidad Distrital Francisco Jose de Caldas

2024

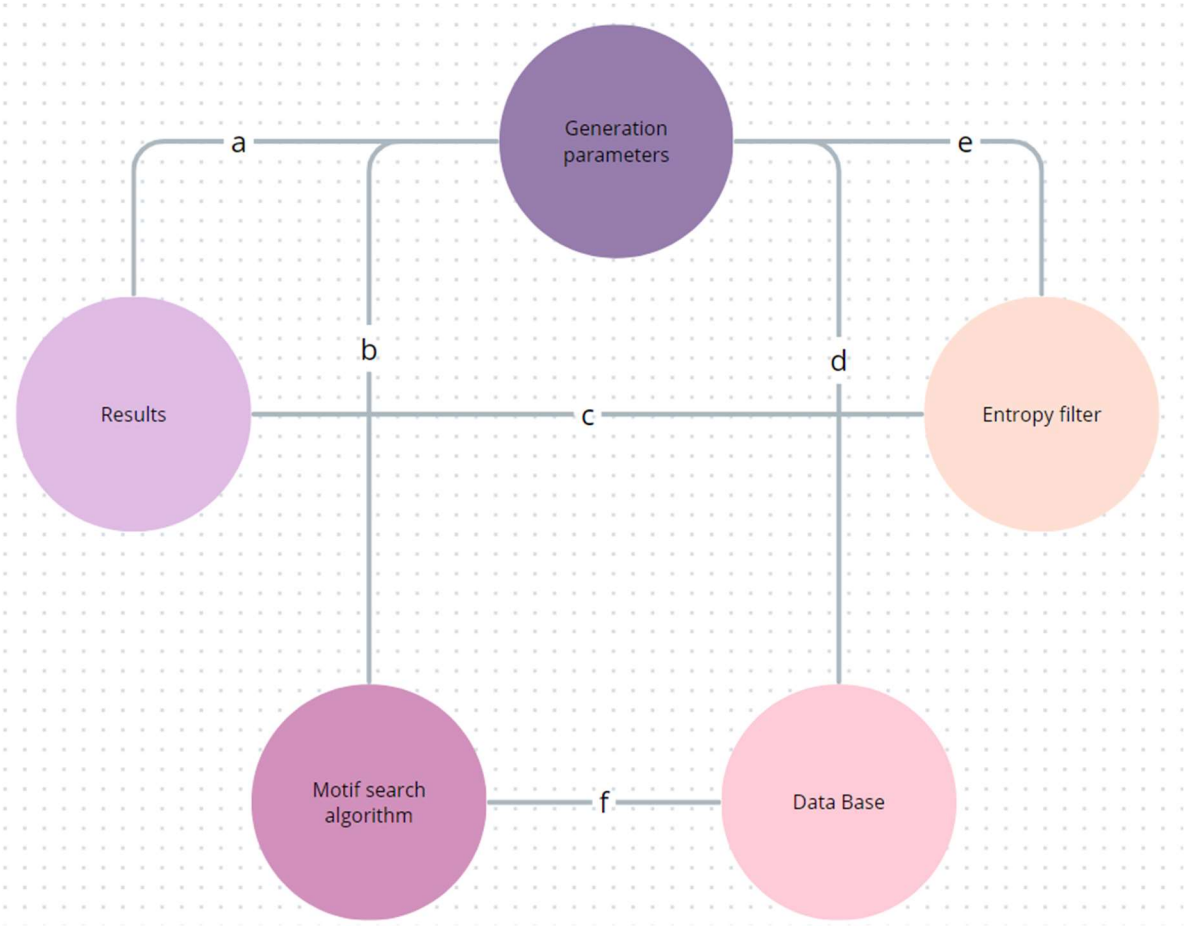
1. Systemic Analysis

A systemic approach can be made from the theory of systems as follows:

1.1. Parts or Elements

- **Database of Sequences:** A collection of artificial genetic sequences generated, composed of nucleotide bases (A, C, G, T).
- **Generation Parameters:** Probabilities of selecting each nucleotide base (A, C, G, T), number of sequences (n), and size of each sequence (m).
- **Motif Search Algorithm:** The process responsible for identifying recurring patterns (motifs) within the sequences.
- **Entropy Filter:** A mechanism to remove sequences with low diversity or repetitive patterns, based on Shannon entropy calculations.
- **Results:** The most frequent motif, its number of occurrences, execution time, and the filtered sequences.

1.2. Relations:



Graph 1

a. **Generation parameters – Results:** Input parameters significantly affect the results obtained.

b. **Generation parameters – Motif search algorithm:** Adjusting these parameters optimizes the search and accuracy and also influences how data is stored in the database.

c. **Entropy filter – Results:** The entropy filter improves quality by removing sequences with low diversity, allowing for more precise motif identification.

d. **Generation parameters – Database:** The parameters ensure that the artificial database effectively simulates realistic genetic sequences for accurate motif analysis.

e. **Generation parameters – Entropy filters:** The parameters of entropy filters are crucial for selecting and ensuring the quality of sequences for motif analysis.

f. **Motif search algorithm – Database:** The search algorithm must adapt to the structure and parameters of the database to effectively identify significant patterns.

1.3. Purpose of the System

The purpose of this system is to identify motifs within genetic sequences to detect recurring patterns that may have biological relevance. These patterns or motifs can indicate functional regions of DNA, such as promoter regions, enhancers, or binding sites for proteins. The system is designed not only to find these patterns but also to ensure that the data used is sufficiently diverse to obtain meaningful results. By filtering sequences with low entropy, the quality of the analysis is improved, ensuring that the detected patterns are relevant.

1.4. Context/Environment

The bioinformatics and computational environment where the system operates is crucial for its design and optimization.

1.4.1. Inputs

- **Probabilities of base selection:** Adjustable parameters that define the proportion of bases A, C, G, T in each sequence.
- **Number of sequences (n):** Determines how many sequences are generated.
- **Size of sequences (m):** Defines the length of each generated sequence.
- **Motif size (s):** Parameter that defines the size of the patterns (motifs) being searched.
- **Entropy threshold:** Defines the limit for filtering repetitive sequences based on their diversity.

1.4.2. Processes

- **Sequence Generation:** Based on the established probabilities for each base, n sequences of size m are generated. This process ensures that each sequence has a controlled but random composition.
- **Motif Search:** The algorithm scans the generated sequences looking for repeated patterns of size s. All possible combinations of bases are evaluated to find the most frequent motif.
- **Entropy Calculation:** Using Shannon entropy, the level of chaos or diversity in each sequence is assessed, eliminating those with low variability that do not contribute to the analysis.

1.4.3. **Outputs**

- **Motif Found:** The most repeated pattern or sequence of size s that appears in the genetic sequences.
- **Number of Occurrences:** How many times the motif appears in the sequence database.
- **Execution Time:** Time required to find the most repeated motif, providing a measure of the algorithm's efficiency.
- **Filtered Sequences:** The database of sequences resulting from applying the entropy filter, removing those with low diversity.

2. Complexity Analysis

The complexity of the problem can be divided into two main parts: **sequence generation** and **motif search**.

2.1. **Sequence Generation**

Generating sequences is a relatively simple task, where fixed-size sequences are created with the bases A, C, G, and T. The time required for this task increases proportionally with the number of sequences and the size of each sequence, meaning that as the database grows, generation becomes slower but remains manageable.

2.2. **Motif Search**

Motif search is more complex because it involves testing all possible base combinations of a given size (motif). As the size of the motif increases, the number of possible combinations grows rapidly, making the search slower. Additionally, the larger the database, the more sequences need to be scanned, which also increases search time.

2.3. **Optimization**

To reduce execution time, techniques such as dividing the problem into smaller parts (divide and conquer) or distributing the task across multiple processes, especially in large databases, are recommended. Using the entropy filter also helps reduce time, as it eliminates less useful sequences.

3. Chaos Analysis

Chaos in genetic sequences is measured using **Shannon entropy**, a formula that measures the uncertainty or disorder in a system. Entropy indicates how varied the sequences are:

$$H(X) = -\sum_{i=1}^{k} P(x_i) \log P\,(x_i)$$

Where $P(x_i)$ is the probability of base x_i (A, C, G, T) appearing in the sequence. High entropy suggests a more diverse sequence, while low entropy indicates many repetitions of the same base.

Entropy-based Filter: To improve motif analysis, we remove sequences with low entropy, as these sequences contain few variations and may skew the results. Setting an entropy threshold allows us to reduce noise and focus the analysis on more chaotic, i.e., more diverse sequences.

4. Results

Results from the experiments are shown here. Tables detail the different configurations of the database, base probabilities, and motifs found. Additionally, the performance of the algorithm before and after applying the entropy filter is compared.

		Probabiliy of Bases							
Sequence size	Database size	A	C	G	T	Motif size	Motif	Motif Occurrences	Time to Find Motif (seconds)
8	1567	25%	25%	25%	25%	4	AGGC	27	0.026239
8	1567	25%	25%	25%	25%	4	AGAC	33	0.0339273
8	1567	25%	25%	25%	25%	4	CATT	23	0.0371078
8	1567	25%	25%	25%	25%	4	CCAT	23	0.0187806
8	1567	30%	30%	10%	30%	4	AACC	40	0.0345455
8	1567	30%	30%	10%	30%	4	TAAA	47	0.0342161
8	1567	10%	25%	35%	30%	4	TGGG	41	0.0415825
8	1567	10%	25%	35%	30%	4	GGGT	56	0.0234785
8	1567	25%	25%	25%	25%	6	TACGGA	3	0.0560987
8	1567	25%	25%	25%	25%	6	TGATCG	2	0.0767575
8	1567	25%	25%	25%	25%	6	CTGGCC	1	0.1039192
8	1567	25%	25%	25%	25%	6	ATTCAC	2	0.1009851
8	1567	30%	30%	10%	30%	6	TTCTCC	1	0.1216517
8	1567	30%	30%	10%	30%	6	CTCAAT	1	0.118049
8	1567	10%	25%	35%	30%	6	GGGTTG	5	0.0820975
8	1567	10%	25%	35%	30%	6	CGTGGG	1	0.1176259

Table 1 Results without entropy filter

		Probabiliy of Bases								
Sequence size	Database size	A	C	G	T	Motif size	Motif	Motif Occurrences	Time to Find Motif (seconds)	Eliminated Sequences(%)
8	1567	25%	25%	25%	25%	4	TCAG	10	0.0074154	39
8	1567	25%	25%	25%	25%	4	CGAT	5	0.0124767	35
8	1567	25%	25%	25%	25%	4	CTGA	7	0.0113226	36
8	1567	25%	25%	25%	25%	4	CATC	6	0.0058289	41
8	1567	30%	30%	10%	30%	4	TCCA	11	0.0062496	54
8	1567	30%	30%	10%	30%	4	TAAC	7	0.0061348	55
8	1567	10%	25%	35%	30%	4	TTGG	3	0.0044124	58
8	1567	10%	25%	35%	30%	4	TGGC	10	0.0040212	55
8	1567	25%	25%	25%	25%	6	GCAAGT	0	0.0287217	36
8	1567	25%	25%	25%	25%	6	TGACCA	0	0.0215566	37
8	1567	25%	25%	25%	25%	6	TCGAAC	0	0.0204159	36
8	1567	25%	25%	25%	25%	6	TGCCCA	0	0.0204907	38
8	1567	30%	30%	10%	30%	6	TCGTAC	0	0.0180661	57
8	1567	30%	30%	10%	30%	6	TTAGAC	0	0.0175735	54
8	1567	10%	25%	35%	30%	6	TGGTCA	0	0.0165447	55
8	1567	10%	25%	35%	30%	6	TCGGGG	0	0.0167881	54

Table 2 Results with entropy filter

5. Discussion of Results

As seen in Table 2, the use of the entropy filter significantly improves execution time by eliminating less diverse sequences. Comparing the tables shows a decrease in motif search time when applying the filter, suggesting that the algorithm is more efficient with more diverse data.

Moreover, chaos analysis using entropy helps reduce noise in sequences, as repetitive sequences do not contribute useful information to motif analysis. However, removing too many sequences could lead to the loss of relevant patterns, so it is crucial to find an appropriate balance in the entropy threshold.

6. Conclusions

• **Optimization through Entropy:** Filtering sequences with low entropy significantly enhances algorithm efficiency and yields more representative outcomes. By focusing on sequences with higher entropy, the system reduces the amount of redundant or less informative data, which in turn allows for a more accurate and faster identification of meaningful motifs. This approach minimizes the computational load and helps in isolating the most relevant patterns within the dataset, thus improving the overall quality of the analysis.

• **Problem Complexity:** The complexity of motif search grows considerably, particularly when dealing with larger motifs. This complexity arises from the need to evaluate a vast number of potential combinations, which can be computationally expensive and time-consuming. To mitigate these challenges, leveraging parallel or distributed processing techniques is crucial. These methods can break down the problem into smaller, more manageable tasks that can be executed simultaneously across multiple processors or machines, thereby reducing the time required to complete the search. Advanced computational resources and techniques, such as cloud computing and high-performance computing clusters, can further enhance the efficiency of the motif search process.

• **Future Optimizations:** To further improve the system's performance, exploring additional distributed computing techniques is recommended. This could involve incorporating more sophisticated algorithms for load balancing and task distribution to optimize resource usage. Additionally, implementing advanced search algorithms, such as search trees, can offer significant benefits. Search trees, such as suffix trees or suffix arrays, can streamline the process of pattern matching and reduce the number of comparisons needed. Integrating machine learning techniques to predict and prioritize potential motifs based on previous findings could also enhance the system's ability to handle large datasets efficiently. Continuous research and development in computational methods and algorithms will be essential to address the growing complexity of motif analysis and to achieve more accurate and faster results.