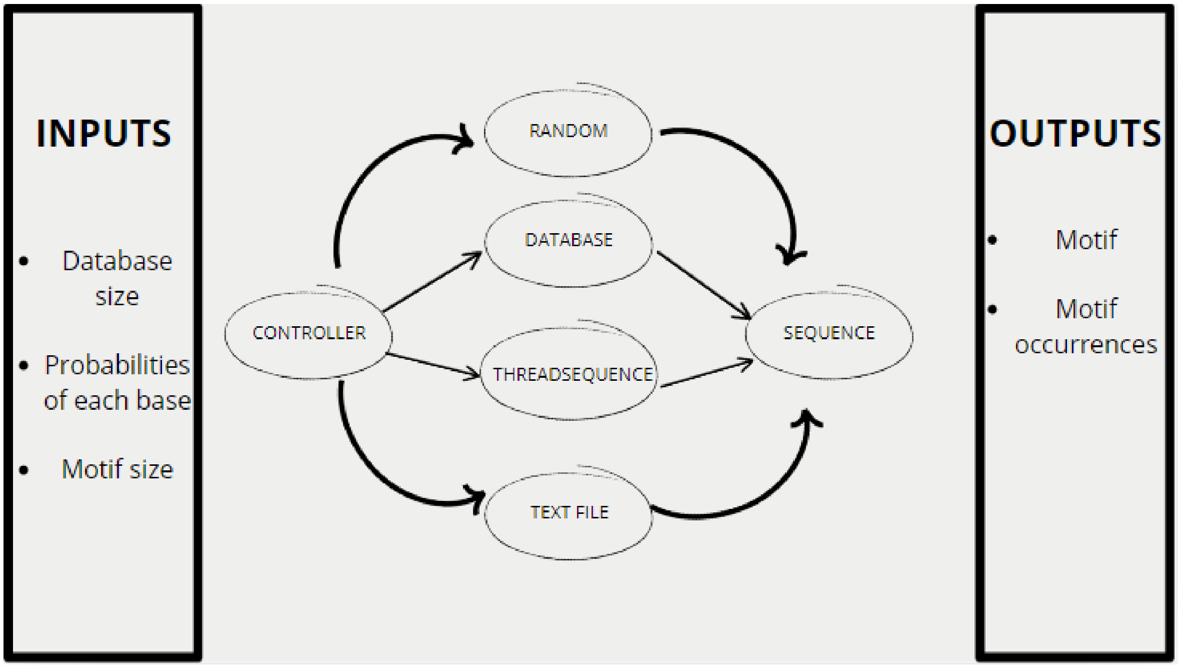
WORKSHOP 1 – REPORT

SYSTEMIC ANALYSIS



Elements:

- Controller
- ThreadSequence
- Text file
- Random
- Database
- Sequence

Relations:

- Controller use Random to generated aleatory numbers that determinate de bases of the sequences.
- Controller create ThreadSequence to divide the creation of the sequences and apply a distributed computing strategy.
- Controller write in the text file all the sequences founded.
- Controller create a database to save the artificial data and from it, the controller search the motif.
- ThreadSequence create sequences to storage it in a database.
- Database storage the sequences.
- Text file save the sequences.

Effects:

- Butterfly: It have a butterfly effect, because when you enter a big number of sequences, the program continue normal, assigned the probabilities and start to create the database, but for the amount of sequences, the data manage could consume all the memory and the program would stop.

COMPLEXITY ANALYSIS

The system's complexity is moderate. The Controller class handles most of the operations, while other elements like ThreadSequence, Database, and Text File interact with sequences. The complexity mainly arises from:

- Multithreading in ThreadSequence to handle large-scale sequence generation efficiently.
- Managing and writing a potentially large volume of sequences to a text file.
- Identifying and counting motifs, which involves iterating through all sequences and motifs.

However, since the Controller manages most of these operations, and the interactions are straightforward, the overall complexity is not excessively high.

CHAOS ANALYS

If you do not filter the sequences, the system will be less chaotic, because it will be easier to predict its behavior, and the system chaos will depend on the probability assigned to each base, because if one base has more percentage of probability, you will have this base in more quantities than the other bases, for this reason between more unbalanced the probability is, less chaos you will have.

However, if you filter the sequences, you will have more chaos, because the sequences will be more difficult to predict, and the big repetitions will be eliminated, and the probabilities will not have to much relevancy in the chaos level than in the other case.

RESULTS

Database Size	Probablility A (%)	Probablility C (%)	Probablility G (%)	Probablility T (%)	Motif size	Motif	Motif Ocurrences	Time to Find Motif (s)
1000	25	25	25	25	3	ACC	407	0,55
10000	15	35	30	20	4	CCC	9824	0,64
100000	30	20	30	20	5	GGAAG	11206	5,07
1000000	20	25	25	30	6	TTTTTT	36863	70,12

In the results it is observed a table with different databases, probabilities of each base, motif sizes, motifs, motif occurrences and time to find motifs, hence it was found that when the sequences or the motif increased, the execution time increased, furthermore, the motif according to the probabilities, it has more or less quantity of each base, because in each experiment, the motif is determinate mainly for the base with more probability. The motif occurrences increase directly proportional to the number of sequences, increasing when the number of sequences increase.

DISCUSSION OF RESULTS

The results highlight that both the number of sequences and the motif size significantly affect execution time. As more sequences are generated, the system handles more data, which slows down processing times. The difference in execution times becomes more noticeable with larger datasets due to increased data management overhead. When fewer sequences are involved, execution times are less variable and more manageable.

The findings also suggest that motif detection is heavily influenced by nucleotide base probabilities. A higher probability of a particular base leads to its dominance in motifs. This, adjusting probabilities can directly impact the most frequent motifs.

CONCLUSION

In conclusion, this workshop demonstrated the application of multithreading for efficient sequence generation and motif detection in genetic sequences. The system is effective for handling large datasets and finding motifs, though performance can be impacted by the volume of data. The chaos analysis showed that filtering sequences increases unpredictability, emphasizing the impact of data filtering on system behavior. Future improvements could involve optimizing data handling and exploring more sophisticated filtering methods to manage large datasets more effectively.