Analysis of the Conservation of Structural Elements in Biological Sequences

Pedro García Suárez

Resumen– El proyecto realizado para este TFG es el estudio del ADN de Elementos Genéticos Móbiles o MGEs de sus siglas en inglés (Mobile Genetic Elements). Estas entidades consisten en material genético encapsulado que es capaz de insertarse dentro de organismos vivos. Los MGEs generalmente se aprovechan de ciertos mecanismos del organismo al que invaden y la finalidad de este estudio es hallar patrones que puedan indicar un comportamiento de este tipo, en concreto en los mecanismos de regulación de la transcripción del ADN. Esto se logrará comparando el ADN de cientos de MGEs con las secuencias de proteínas que se sintetizan en los organismos que estos infectan. Las coincidencias que se encuentren serán contextualizadas utilizando casos de control para extraer el máximo de información útil posible de ellas. El análisis se realizará mediante varios scripts en lenguaje Python.

Palabras Clave– Bioinformática, MGEs, Python, BioPython, Proteínas, Motivos, Regiones Promotoras, Factores de Transcripción

Abstract– The project executed for this TFG is the study of the DNA in Mobile Genetic Elements (MGEs). These entities consist of encapsulated genetic material that can insert itself in a live organism. Generally, MGEs take advantage of certain mechanisms of the host and the goal of this study is to find patterns that may indicate this type of behaviour, specifically the mechanisms that regulate the transcription of DNA. This will be achieved by comparing the DNA of hundreds of MGEs with the sequences of proteins that are synthesized by the organisms infected. The matches found will be contextualized using control cases to extract the maximum amount of useful information about them. This analysis will be implemented by different scripts written in Python.

Keywords– Bioinformatics, MGEs, Python, BioPython, Proteins, Motifs, Promoting Regions, Transcription Factors

1 Introduction - Context of the project

MOBILE genetic elements (MGE) [1] are basic units of DNA contained inside a capsid that are able to move inside an organism such as the human body.

The term MGE refers to numerous, very different types of entities such as viruses, plasmids or phages. But one thing that they have in common is that they all depend in a lesser or greater extent on a host that provides cell machinery so that they can replicate and move around.

A classic example is the case of viruses. A virus introduces its genetic material inside a host cell when it infects it. After that, these genes start giving instructions to the host cell proteins and enzymes such as RNA and DNA polymerases to replicate the virus using the host's resources. Once the replications are made and there is no room for more, the cell explodes and sets them free to infect other cells.

Nevertheless, MGEs are believed to being able to use more systems of the cell for their own purposes. Specifically, they can use the transcriptional regulatory networks (TRN) of the cell. These networks operate activating and repressing the transcription of certain genes and therefore blocking or promoting the use of the genetic information that they may contain.

TRNs work using transcription factor (TF) [3] proteins that can bind to the DNA molecule at specific locations by recognizing a pattern in the molecule called sequence motif.

• E-mail de contacte: pedro.garciasu@autonoma.cat
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Joan Serra Sagrista (DEIC)
• Curs 2021/22

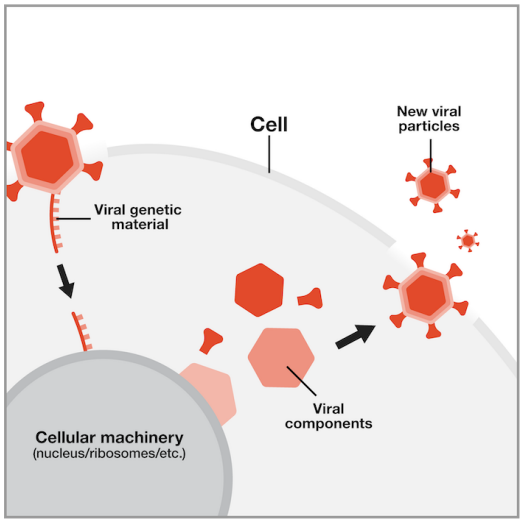

Fig. 1: Virology Infection [2].

The specific location is the region that promotes the gene of interest and the TF is the one that represses or promotes the activity in that area of the RNA-polymerase holoenzyme, the protein that actually transcripts the genetic information of the DNA.

1.1 State of the Art

An example of MGEs taking advantage of host TRNs can be found in the use of the host SOS response [4] to damage in the DNA. In normal conditions the genes capable of repairing the DNA are tightly repressed by TF LexA [5]. Some of these genes are capable of killing the host so it is important that they do not get promoted when it is not needed.

Once DNA damage is detected, LexA lets these repair genes off of its grip making them able to undergo transcription and help in the repair tasks. Reasonably, MGEs would also like to react when damage in the DNA is detected and they do that by using LexA binding sites in some parts of their genetic material, usually the ones in charge of destroying the cell. This means that when LexA is deactivated by the SOS response, it sets the MGE's genes free to express themselves and kill the cell to finally abandon it for a new host.

1.2 Objectives

The SOS response co-opting is one example of host TRNs being used by MGEs but it is believed that other transcription networks can be co-opted by the MGEs as well. In this project we will analyze the genetic material of MGEs generating data that then can be studied to determine if a region of a DNA sequence can be considered suspect of taking advantage of the mechanisms of the cell.

To do that we will collate both MGEs and host cells proteins to find matches of binding sites on MGEs with the protein motifs of the host cell, which in essence are patterns of nucleotides found in the DNA chain that can be compared to find the most probable position in which a protein can operate.

In other words, there already exist cases of MGEs taking advantage of transcription mechanisms of the cell like the SOS response co-opting. With this dissertation we aim at finding other cases of this phenomena, or at least generating a rich statistical analysis of how potential binding sites are located in the DNA of MGEs. Our results will try to show the distribution of binding sites for many combinations of MGEs+Proteins and see if any interesting conclusions can be drawn from them.

In order to achieve this goal, we will need to take smaller steps to get to where we want:

- First, we need to elaborate a way to find matches between DNA sequences and protein motifs.
- Once we can do that, we may focus on running as many cases as we can and produce a statistical data of how the matches are distributed in the DNA of different MGEs.

In the first section we will also need to find a consistent structure to store all the input data (MGEs sequence files and protein motifs) in a way that is accessible by our script and find a good option to store the output data to facilitate the following statistical analysis.

The second phase will consist of computing different statistics from the results of the first phase and elaborate a reliable control case that makes the findings that we make valuable and meaningful. We will also normalize the results for a better comparison between cases.

2 Methodology

The working process for this project has consisted of identifying the tasks and perform them on a weekly basis. We can identify this methodology as a waterfall methodology. We think this model adapts the best to our time requirements and also our capabilities. Other options were considered but the uncertainty of some aspects of the project made us decide towards a traditional way of doing things, such as the waterfall model.

Each week a Lab Meeting with the tutor has been conducted to assess the progress made in the project and also guide the work flow into the preferred course. We would also debate about the different solutions that could be used and find the option that fitted best for every task.

These meetings have helped with comprehension of the matter of this project serving sometimes as lectures on the different elements that we work with, including the mechanisms of the cell, tips about coding in BioPython, where to search for reliable information and how to evaluate the results that our code returns.

The nature of this methodology probably made the planning of the project somewhat unclear at the beginning, where the tasks were defined vaguely and did not cover the whole extent of this project. The reason behind this can be related to the small number of meetings conducted prior to that moment which meant having less information, or at least not as specific, about the goals and steps to follow in this project.

This methodology has worked fine and has made it easy to correct the path followed in development on a weekly basis. Therefore, it has continued to be the preferred methodology, adding sporadic questions and doubts which the tutor has tried to solve as quickly as possible.

Following, we will go over the whole the realization of the project to show the progress made towards accomplishing the goals stated on the introduction of this document:

2.1 Biology Lessons

One big drawback at the start of this project was the reduced knowledge we had about the field of study. Even though we were familiar with basic concepts like DNA and cell replication, there was still a vast number of concepts that escaped our reach and that needed to be understood.

The level of understanding was not too demanding but it was enough so that the first two weeks were destined to getting familiar with the transcription process inside cells and how MGEs may take advantage of them. This knowledge will be crucial to understand what we are doing and to know where to look when we analyze the results.

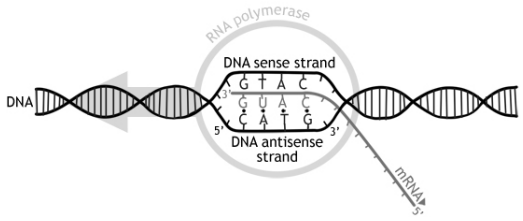

Fig. 2: Simplified depiction of DNA transcription [6]

We also took this time to read about the different ways of representing biological data for computing. From the files that contain the entire DNA sequence of an organism to the structures that represent a protein motif and the tools that exist to work with this type of information in bioinformatics.

2.2 BioPython

The next step in our project is to learn how to use BioPython correctly. Specifically, we are looking at the functionalities that best fit our project.

BioPython is a library of Python that includes functions used to manage data structures related to bioinformatics. In this project, the most important aspects for which we use this library are to parse the content of protein and DNA files. SeqIO is the function that lets us treat this type of data files and convert their content into structures that we can work with.

Luckily, we had at our disposal sample code [7] with functionalities similar to the ones that we needed to code. Specially those related to the calculation of the score of the DNA sequences using the pssm [8].

2.3 Initial Script

We approached the elaboration of the script with the idea of building the main functionality of the code first and later adapting it to read multiple files and store results.

The main functionality of the code is to process a protein motif and the DNA sequence of an MGE and then find matches that can be considered potential binding sites for transcription factors.

The algorithm on one hand creates a motif from a file using SeqIO. This motif is an object in BioPyhton that contains all the information about a transcription factor including the frequency of each nucleotide in the sequence and their position. It then generates a Position-specific Scoring Matrix (PSSM) which is another way to represent this information and finally calculates a threshold that will act as a filter for the hits with a high score from the hits with a lower score.

The matrix itself has a function where you pass a genetic sequence and it returns the score for each position of the sequence. This score represents the probability of a binding site starting in that position in the sequence. In other words, if the nucleotides of a segment of the sequence are complementary to the sequence of a transcription factor.

On the other hand we read a genome file and get the DNA sequence of the MGE into a list format. When we have all the data structures we need, we just run the PSSM through the DNA sequence and calculate the score of each position of the DNA, saving those with a higher score than the threshold.

After we end up with a list of hits, we check for hits that may overlap one another, that is, that are separated from each other by less than half the length of the motif. When we find hits that overlap one another, we keep the one with a higher score and delete the other one. This is done because overlapping is just a consequence of using the probabilities of a hit being located in a region.

For example, in a region with a really high score, we will hits really close to each other because if you change just one nucleotide of a hit, it will still have a high score and be considered a match and also represent the same region. Deleting the overlapping hits will mitigate that and leave the highest hit to represent that region.

Finally, when we have the list of potential hits, we sort them out by their relative position to the closest gene in the genome. The motivation for this is that the most interesting hits will be the ones located in the promoter regions of genes, we will say that these hits are in the operator region.

Generally, these regions are the ones just before the start of a gene. For this project we are calling operators those hits located between 250 base pairs upstream of the start of a gene and 50 base pairs downstream of the start of a gene. The other cases will be classified as inter-genetic if they are outside of a gene and intra-genetic if they are inside of a gene.

Naturally, this last step can only be done if the genome file we are analyzing contains information about the genes coded in its sequence. Some do not contain this information and therefore we can not know where a gene starts and finishes.

2.4 Adapting the Script

Once the main functionality is finished, we need to automate the process of feeding the code with the files that it needs to run. At this stage, some decisions were made in terms of which organisms we would be focusing on for the study, given that our scope is limited by the available databases of genomes and transcription factors.

Taking all this into consideration, we decided to use the database of phages from Millard Lab for the DNA

sequences since it has a very high number of specimens (around 16.000) and for most of them it specifies the host organism where the phage lives. This will let us filter phages by the genus of a host of interest, for example Bacillales.

This is interesting because if we know the host that our MGEs infect, we can focus on proteins that this specif host synthesises and look if our MGE has parts of its genetic sequence similar to that of the protein. That would mean that the MGE is trying to blend in and use mechanisms of the host cell.

Even though this script is prepared to work with different motif files, we will only use the LexA protein file because real cases of binding sites have already been found with this protein. In this study, we will search for binding sites of

Fig. 3: Weblogo [9] of LexA in Bacillales.

LexA in the DNA of phages that infect organisms that belong to the order of Bacillales. The program takes around 8 minutes to execute completely due to the high number of genomes that it has to analyze (898). The bottleneck of this script is the function that calculates the score of the DNA sequences. It needs to loop over DNA chains containing thousands of elements and calculate the score for each of them.

The process to filter the phages by their order is to get a list of every descendent of the clade of interest (Bacillales) through an API of the National Center for Biotechnology Information (NCBI). Once we know all the organisms under Bacillales, we look on Millard Lab's database for phages that infect hosts in this list. The database specifies the accession number of the genomes which is used to download the files containing the DNA sequences.

3 Preliminary Results

The first results we get are all the hits of protein LexA that we can find in the genomes of all the phages of our analysis. We will save as much information as possible because it may be useful in the post analysis. This includes information about the closest gene to a hit, the score of the hit and its location.

The reason behind breaking down the analysis in two steps is to separate the time-heavy task of finding all the hits and the second phase which is much quicker in terms of computing time. That way, we can produce the hit results first and then focus on classifying them by organism.

The preliminary results do not say much if presented plainly. They need treatment and contextualization. They also need to be confronted with a control case to see which parameters might be interesting and should get our attention.

The idea is for this control case to be the multiple permutations of the motif of LexA. This means, changing its sequence randomly multiple times and then executing the script again with each permutation. This way we can compare the real data with data produced by a random motif and see if the real data is actually saying something.

Nevertheless, some promising information can be extracted just from the plain results. There are some hits that stand out for being in tandem. That means that they are really close to each other in the same promoter region. That is a great sign that can mean that there is a binding site in that region. We will be looking at the distribution of these elements to see if our analysis is useful in some way.

3.1 Analysis of output

The next and final step of this project is to write a script that takes in the file with all the hits and classifies them by phage and also computes certain parameters of interest.

First it will extract some statistical information from the results themselves. This includes:

- GC content of the genome at question: This is a metric tosee the distribution of nucleotides in the genome. If this number is high, it means the genome contains a high number of nucleotides C and G and therefore it will have a higher probability of producing matches with motifs that contain a high quantity of C and G.
- Number of hits classified by their region (Operator, Intragenetic and Intergenetic).
- Average score of the hits classified by region.
- Average sum of normalized scores (by region): This will be useful to identify those phages with high score hits.

We will treat the raw results using the pandas python library. It is a useful way to extract information from csv files and convert their content to pandas dataframes. Dataframes are very malleable and fast to operate with. They are helpful when calculating averages and classifying the results by region.

The final conclusions drawn from the results will be made using Excel to produce graphs that show the tendencies of interest

4 Results

When having such a big collection of data points, it is crucial to know where to look for patterns that may corroborate or deny any given hypothesis. A good method when available, is to have one case in the sample that are confirmed to follow the hypothesis and other cases generated randomly, this way you can look at the confirmed case when analyzing different metrics to see how it responds relative to the random ones.

If the real confirmed case reacts in a different manner than the random fake cases, that may be a metric to turn our attention to since we are observing that a real case reacts to it and the false do not. This means that if other real data points react similarly to the confirmed case, we can consider them potential cases that follow our hypothesis.

The method that we have used to analyze our results has followed this policy. Our confirmed data point will be the phage pGIL [10] that is known for taking advantage of the

SOS response by using LexA binding sites. For the random samples, as explained before, we will use permutations of the motif of LexA to generate new 'fake' hit results.

We will focus on the stats for hits found in the operator region, since this is the region where a real hit could be found. The case that intragenetic regions could also be potential binding sites admits discussion but escapes the scope of this project.

The metrics that have shown better results for the real confirmed case and bad results for the false samples are the ratios of the sum of normalized scores in the real hits over the false hits for each phage. If a phage has a ratio over 0.5, it means that the hits found in this phage have higher scores with the real protein than with the randomly generated ones. If it is under 0.5, it means that the hits with the real protein do not stand out from the ones with the random protein. We have selected this metric because it shows a good ratio for the known cases of pGIL phage (related to the SOS response).

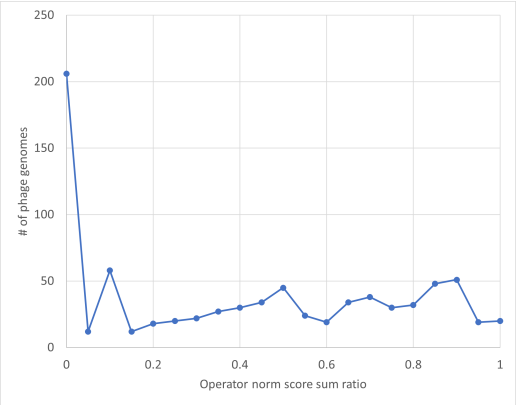

Fig. 4: Number of phages over their ratio.

Figure 4 shows that the majority of phages have hits with a low ratio as expected. The norm is that phages do not have a lot of binding sites for proteins of the host, so the majority of them have hits with a low ratio which means that their hits are more likely to be false hits.

Finally we will look at an independent metric from the one seen above, that is the number of tandems. The hypothesis is that only potential binding sites will have tandems, so there should be a low number of phages that contain tandems.

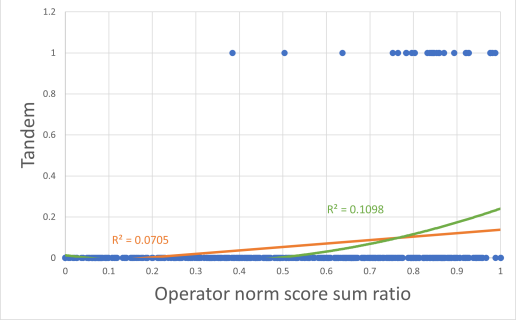

Fig. 5: Distribution of tandems.

As expected, figure 5 shows the distribution of tandems by the ratio of the phages. First, we confirm that tandems are rare events and second, that tandems occur more frequently (green distribution) in phages with a higher ratio, which means that tandems could be a valid metric to identify potential binding sites. Once again we remember that the ratio means how much the hits of a phage with the real protein stand out from the hits of the same phage with a fake protein. So having more tandems when the ratio is higher shows tandems as an independent metric with similar results.

5 Conclusions

These results show that the study of binding sites using the motif matrix to calculate scores can be considered a promising option that should be explored. Lab teams around the world are investigating this method to see if it is useful. If this method shows consistent results and also finds new cases like the SOS response, it could be revolutionary since the fact that MGEs can take advantage of the cells transcription networks frequently opens new opportunities in the fields of biology, pharmacology and medicine.

Imagine a genetically engineered MGE that can take advantage of certain mechanisms of the cell. If we tell it what to do, it could for example suppress certain harmful mechanisms or provoke cell destruction in cells of our choice, like carcinogenic cells.

Nevertheless, the results shown in this project could be more refined and there are still areas to improve, like saving the matrix of the permutations to see if they are randomized enough to be considered control cases. Also, we found some metrics that support our hypothesis but there can be more, better ones still to be found. This project could be continued by studying more combinations of proteins and phages, giving a bigger sample to draw conclusions from. The number of permutations of the motif could also be incremented to eliminate as many false positives as possible.

Thanks

I would like to thank:

Ivan Erill, for his help throughout the elaboration of this project and for his commentaries.

Joan Serra, for his commentaries along the project that helped improve in the elaboration of reports and this memory.

My friends and family for their support.

UAB and Escola d'Enginyeria for the knowledge and skill that I have learned there and that I have used for the elaboration of this project.

References

[1] Wikipedia. (2021, September) Mobile genetic elements. [Online]. Available: https://en.wikipedia.org/wiki/Mobile_genetic_elements

[2] N. H. G. R. Institute. (2021, November) Genomics and virology. [Online]. Available: https://www.genome.gov/about-genomics/fact-sheets/Genomics-and-Virology

[3] I. Erill. (2006, February) Information content in transcription factor binding sites. [Online].

Available: https://erilllab.umbc.edu/files/2016/04/Introduction_Information_Theory.pdf

[4] Wikipedia. (2020, December) Sos response. [Online]. Available: https://en.wikipedia.org/wiki/SOS_response

[5] M. B. Nadine Fornelos, Douglas F. Browning. (2016, March) The use and abuse of lexa by mobile genetic elements. [Online]. Available: https://doi.org/10.1016/j.tim.2016.02.009

[6] Dovelike. (2009, January) Dna transcription. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/f/f7/DNA_transcription.jpg

[7] A. Kennedy. (2021, July) Pssm model. [Online]. Available: https://github.com/ErillLab/cgb/blob/master/cgb/pssm_model.py

[8] Wikipedia. (2021, January) Position weight matrix. [Online]. Available: https://en.wikipedia.org/wiki/Mobile_genetic_elements

[9] I. Erill. (2009, April) What is (in) a sequence logo? [Online]. Available: https://erilllab.umbc.edu/files/2016/04/Introduction_Information_Theory.pdf

[10] N. A. Caveney, A. Pavlin, G. Caballero, M. Bahun, V. Hodnik, L. de Castro, N. Fornelos, M. Butala, and N. C. Strynadka, "Structural insights into bacteriophage gil01 gp7 inhibition of host lexa repressor," *Structure*, vol. 27, no. 7, pp. 1094–1102.e4, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969212619301157