DNA CODIFIER
A Bioinformatic Web Platform

Maria Eduarda Vanin Miotto[1]

[1]Área de Conhecimento de Ciências Exatas e Engenharias -
Trabalho de Conclusão de Curso -
Curso de Sistemas de Informação -
Universidade de Caxias do Sul (UCS)

mevmiotto@ucs.br

Abstract. *The benefits arising from the advancement of technology and the development of systems in recent decades have enabled the evolution of access to information. Among the favored areas that demand the growth of their analysis, there is molecular biology. The creation of a web platform that facilitates operations and enables the interaction of nucleotide sequences could provide a breakthrough for researchers in the area. Based on this demand, it is necessary to draw up activity diagrams, use cases and tests to determine the best alternative to develop a system that meets the established functions and requirements. The use of literature made it possible to analyze the choice of the most adequate and efficient technologies for the construction of the project. Furthermore, it is evident that researches in the area of bioinformatics must continue to expand, given its complexity and the need to achieve a better understanding of genetic information.*

1. Introduction

Organisms are made of cells composed of genetic material. Its sequencing results in strings composed by nucleotides. These nucleotides are adenine, cytosine, guanine, thymine and uracil, which are characterized by the letters A, C, G, T and U. The first four nucleotides are part of the formation of the deoxyribonucleic acid (DNA). The last one, uracil, is only found on the ribonucleic acid (RNA), replacing the thymine. Since the sequences are long and complex to be analyzed without a computer, it became necessary the influence of a diversity of algorithms to analyze and distinguish properties of the molecules and study its order [Pevsner 2015]. Bioinformatics researchers with access to web systems that provide tools for their areas, can bring large benefits for the genome studies [Veloso et al. 2015].

Algorithms are sequences of instructions that are comprehended by computers. It becomes functional when written in a programming language. Depending on the purpose of the project, it can be necessary to use front-end technologies, back-end or in most cases both combined [Sebesta 2003]. Developing a web application even without a database, requires front elements such as HTML for the basic code syntax, CSS for code style and Javascript for creating the functions for web. The back-end is required for the manipulation of the text files uploaded followed by the script (function) selected (which were all written in Python) besides the use of the Flask framework.

The growth of technological resources along with the expansion of data available, made the development of applications, programs, services and tools related to the molecular biology area, something possible to achieve. Rost, Yachdav and Liu analyze sequences, predict proteins and related operations through a web service where the user must inform data by an input and the system executes the function when clicked on the submit button [Rost et al. 2004]. Diniz and Canduri report in their paper a cycle that represents biology systems. Along with every discovery in the area, new hypotheses are raised, and biological aspects are questioned. Furthermore, comes the necessity of creation of new softwares to supply the demand [Diniz and Canduri 2017].

There are physical features that could be examined to achieve a better understanding of the genome, them being the entropy, enthalpy, base-pair stacking and stability [Martinez 2018]. The properties selected to be analysed are based on their big relevance on promoter regions characterizations [Benham 1996]. These parameters were presented in previous projects and can prove that a DNA sequence when represented in numeric attributes instead of nucleotides, results in a big gain of information [Bansal et al. 2014].

The implementation of the DNA Codifier platform has as the main objective to be attended, the development of an interface available online that receives DNA and RNA sequences and converts them into numeric values, where students, researchers and any person interested in the bioinformatics area could have access.

2. Material and Methods

The process of development for the DNA Codifier platform implied the use of a web server that would allow the files uploaded by the user to be manipulated and available for download, besides the support of the framework chosen on the project, being in this case the Flask Framework. Pycharm Professional was chosen to provide the connection between the front-end and back-end of the application, allowing effectiveness and fast connection for the resources. The front-end was implemented with HTML (HyperText Markup Language), CSS (Cascading Style Sheets), Bootstrap Framework and Javascript, while the back-end was implemented using Python language and Flask Web Framework.

Despite being a micro framework, flask allows applications to be built by an Model-View-Controller (MVC) architecture. The model stores the data and interacts with the controller. The controller accepts the user inputs and enables the connection between the view and model, being split into initialization, routing and execution. The view renders data from the application, storing the html files in a templates folder and the css and javascript files in a static folder. As new applications are constantly being developed in flask, it is becoming useful to adoption use the Model-View-Controller structure (as can be detailed in Figure 1), which brings satisfactory results for bigger and more complex projects.

The application presents five distinguished functions which were all written in Python language by students from the bioinformatics area of UCS. It works when a user inputs at least one nucleotide sequence into the text area or loads a .FASTA or .txt file. The results will be available for download on .txt and .csv extensions. The first task shuffles the elements from a sequence, and the next four tasks calculate the enthalpy, entropy, stability, and stacking between the nucleotides of a sequence, generating for the user the numeric values of these properties.

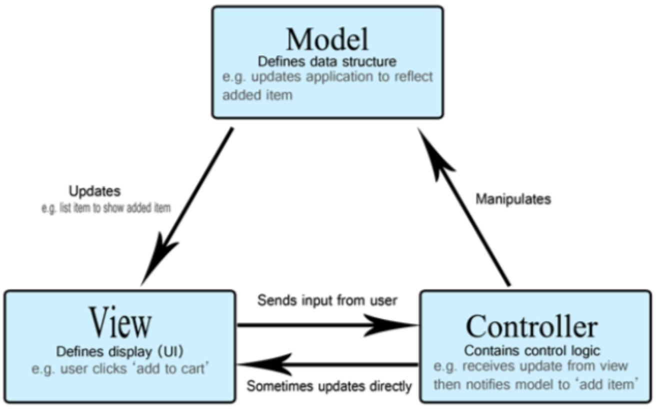
Model
Defines data structure
e.g. updates application to reflect added item
Updates
e.g. list item to show added item
Manipulates
View
Defines display (UI)
e.g. user clicks 'add to cart'
Sends input from user
Sometimes updates directly
Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'

Figure 1. The relation between Model, View and Controller

3. Results

The bioinformatics web platform was developed to be available for any operating systems compatible with the web browsers Google Chrome, Safari, Firefox, Opera or Microsoft Edge. According to the use cases (Figure 2), the user is the only author and it can execute five functions, them being: i) shuffle sequence ii) calculate entropy iii) calculate enthalpy iv) calculate stability v) calculate stacking.

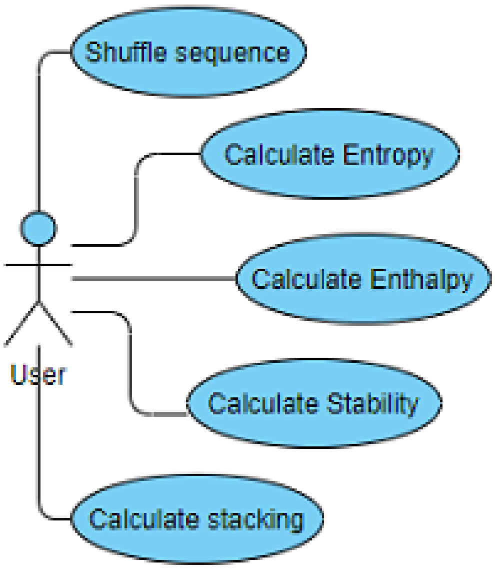

Figure 2. Diagram of Use Case

According to Figure 3 and the definition of activity diagrams, it expresses the stages of the process. It's determined that the user selects the function of choice and in advance, enters the sequence(s) by text or by inputting a file. Following the process, the system analyses the data received and determines if it's valid or invalid. The results should be able to download the instant the values are generated and the process is finished.

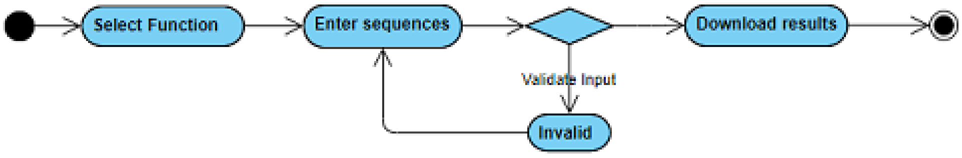

Figure 3. Diagram of Activities

The interface structure works following a sequence of steps, organized in an easy and understandable approach. The elements on the page only become available when the previous step is completed. The user can only provide a nucleotide sequence or a file, when the function has been selected, and only after that, the "get results" button is available. In this condition, the download buttons only become available when the "get results" button is clicked. The platform also allows an accessible tool that informs the user details about how the data should be informed, what it's acceptable and recommended.

The implementation of test cases indicates which aspects of the interface should be specified, according to the elements names and their justification. The first line of the tables include the test case, which means the function name of the application. The Table 1 only presents the "shuffler" function, while the Table 2 presents the other four functions "Enthalpy", "Entropy", "Stability", and "Stacking". The description field identifies the objective of the operation selected. The preconditions describe the conditions that must be satisfied to achieve the expected result. The inputs inform the parameters of entry, while the outputs inform what should be returned for any object that calls the operation. Lastly, the post conditions describe the expected status of the component.

Table 1. Test cases - Shuffler

Test Case	Shuffler
Description	The function will receive one or more nucleotide sequences and will shuffle the elements.
Preconditions	One or more nucleotides sequences composed by the letters A, T, G, C and U (RNA).
Inputs	The entrance must have at least two nucleotides.
Outputs	The number of elements must be equal to the number of input elements.
Post Conditions	Nucleotide sequence in a different order from input.

The DNACodifier web application was developed based on a single Flask module, since it wasn't necessary a database for the project, the architecture of the project follows the flask structure shown in the code below. The app.py is a file written in python language which contains the functions that start the application, it allows the entries on functions and terminates the processes. The config.app file is also written in python and includes the configuration of the variables and the content of the structure. The requirements.txt is a text file that contains all the package dependencies for the project. The static folder has the purpose of storing content that cannot be changed with a user's action. The styles in css format and functions on javascript language defined for the applications, are stored

Table 2. Test cases - Enthalpy, Entropy, Stability and Stacking

Test Case	Enthalpy; Entropy; Stability; Stacking
Description	These functions will calculate the numeric attributes of the nucleotides referring to their properties.
Preconditions	One or more nucleotides sequences composed by the letters A, T, G, C and U (RNA).
Inputs	The entrance must have at least two nucleotides.
Outputs	The system will return a file with negative numeric values, always being the number of elements from the inputs less 1.
Post Conditions	The numeric values need to be equivalent to the properties of the elements entered.

on their own folders, css and javascript that are inside the static main folder. The files are named style.css and function.js, and they all support the web pages. The folder named templates is a default directory that embraces all the html files in any project, in this case being the files index.html, selection.html and result.html. The code below presents the structure of organization of the project files:

```
app.py
config.py
requirements.txt
static/
    css/
         style.css
    javascript/
         function.js
templates/
    index.html
    selection.html
    results.html
```

4. Discussion

A free website can lead to benefits for science as it becomes available for a bigger public. However even with the evolution of technology, algorithms and computer access, little has been made about the semantic integration with biological data [Cannata et al. 2008]. Algorithms in bioinformatics and systems of biology are separated by a bridge, where methods would be necessary to create an intersection between these paths [Francesconi et al. 2019]. Bioinformatics applications must contribute with the organization of data for a better access of information and creation of entries and results, helping data analysis and interpretation [Luscombe et al. 2001].

Due to the evolution of application development, not only back-end technologies are essential because of their security, integrity and functionalities, but also an user platform interface is necessary to allow through front-end resources a better navigation and efficient design components [Newman et al. 2018]. Aspects such as the data storage capacity, the ease of access, the processing speed and the integration of multiple tools also interfere in the user experience when accessing a platform [Catanho and de Miranda e Wim Degrave 2007].

Analyzing the DNA Codifier and getting based on the results obtained until the present moment, some similarities can be found between the application and a platform released in 1992 known as Predict Protein (https://predictprotein.org/). This application receives as an input a protein sequence, and outputs multiple sequence alignments.

Over the years, more organisms are being sequenced and available for the scientific community. With that being stated, the necessity of working with that data in an efficient way grows. The parametrization of genetic sequences in structural features can result in information gain. There are more than 120 features that hold physical and structural information from a DNA molecule. Therefore, the development of this application can generate data that can contribute with highy gain for the community [Martinez et al. 2021].

5. Conclusion

Molecular biology is considered a wide area that needs to be constantly explored. The agile evolution of technology strongly contributes to the advancement of its studies, providing new possibilities and achieving unprecedented results. The creation of capable algorithms that process properties of a large volume of DNA and RNA sequences in a short time period is a strong example of the advantages of growing machine processing capacity. Thus, the integration of algorithms to open access web platforms can bring many benefits to research progress.

As presented on TCC I and in the writing of this article, planning for the development of a system depends on many factors. The results were based on literature of similar projects, analyses of articles, cases of test and creations of diagrams. Planning on works to be made in the future, the platform could be improved, so the data entry could be amplified, accepting not only nucleotide sequences but also amino acids.

References

Bansal, M., Kumar, A., and Yella, V. R. (2014). Role of dna sequence based structural features of promoters in transcription initiation and gene expression. *Current Opinion in Structural Biology*, 25:77–85. Theory and simulation / Macromolecular machines.

Benham, C. J. (1996). Duplex destabilization in superhelical dna is predicted to occur at specific transcriptional regulatory regions. *J. Mol. Biol.*, pages 425–434.

Cannata, N., Schroeder, M., Marangoni, R., and Romano, P. (2008). A semantic web for bioinformatics: Goals, tools, systems, applications. *BMC bioinformatics*, 9 Suppl 4:S1.

Catanho, M. P. and de Miranda e Wim Degrave, A. (2007). Comparando genomas: bancos de dados e ferramentas computacionais para a análise comparativa de genomas procarióticos. *Revista Eletrônica de Comunicação, Informação e Inovação em Saúde*, 1(2).

Diniz, W. and Canduri, F. (2017). Review-article bioinformatics: an overview and its applications.

Francesconi, M., Di Stefano, B., Berenguer, C., de Andrés-Aguayo, L., Plana-Carmona, M., Mendez-Lago, M., Guillaumet-Adkins, A., Rodriguez-Esteban, G., Gut, M., Gut, I. G., Heyn, H., Lehner, B., and Graf, T. (2019). Single cell rna-seq identifies the origins of heterogeneity in efficient cell transdifferentiation and reprogramming. *Elife*.

Luscombe, N., Greenbaum, D., and Gerstein, M. (2001). What is bioinformatics? a proposed definition and overview of the field. *Methods Inf Med*.

Martinez, G. S. (2018). Promoter sequence characterization through the analysis of enthalpy, entropy, stability and base-pair stacking values.

Martinez, G. S., Sarkar, S., Kumar, A., Pérez-Rueda, E., and de Avila e Silva, S. (2021). Characterization of promoters in archaeal genomes based on dna structural parameters. *MicrobiologyOpen*, 10(5):e1230.

Newman, C., Agioutantis, Z., and Schaefer, N. (2018). Development of a web-platform for mining applications. *International Journal of Mining Science and Technology*, 28(1):95–99. Special issue on ground control in mining in 2017.

Pevsner, J. (2015). *Bioinformatics and functional genomics*.

Rost, B., Liu, J., and Yachdav, G. (2004). The predictprotein server. *Nucleic Acids Res*.

Sebesta, R. W. (2003). *Concepts of Programming Languages*. Addison-Wesley Longman Publishing Co., Inc., USA, 6 edition.

Veloso, H., Vialle, R., and Ortega, M. (2015). *BOWS (bioinformatics open web services) to centralize bioinformatics tools in web services*.