**MORGAN HUNTER TOOL – USER MANUAL**

The Morgan Hunter Tool is a Python-based computational tool that searches for motifs on DNA sequences using consensus sequences represented as Position Specific Frequency Matrices. The program provides some of the results in .csv files, others as graphical representations of the analyzed sequence(s), and yet others are displayed on screen.

The Morgan Hunter Tool is divided in three modules:

* Morgan Hunter (named after Sir Henry Morgan, the navigator and Dr. Thomas Hunt Morgan, the geneticist) searches for motif occurrences along DNA input sequences.
* Morgan Plotter produces the graphical representation of the analyzed sequences, with the occurrence of each motif found and the positions of satellite sequences (these must be provided by the user). The similarity of each detected motif with the consensus sequence is color coded. Morgan Plotter uses Morgan Hunter’s output information, and therefore, these modules must be run sequentially.
* Morgan Score is used when the motif/consensus similarity of large amounts of sequences is to be found. It does not provide a graphical representation of the sequences. Morgan Score uses Morgan Hunter’s output information, and therefore, these modules must be run sequentially.

**RUNNING THE MORGAN HUNTER TOOL**

Download Anaconda Distribution (<https://www.anaconda.com/distribution/>). Make sure that the folder containing the three Morgan Hunter Tool modules (Morgan\_Hunter.ipynb, Morgan\_Plotter.ipynb, and Morgan\_Score.ipynb) also contains the following items (Fig. 1):

Imagen que contiene captura de pantalla, texto

Descripción generada automáticamente

Figure

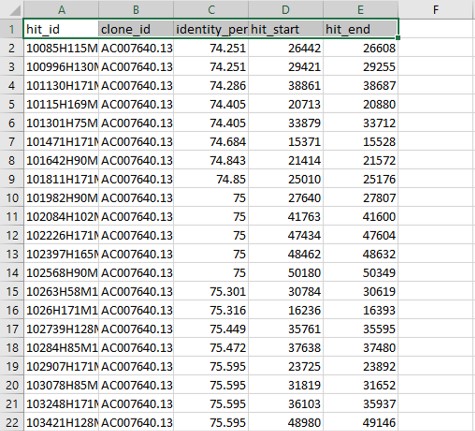
* Morgan Hunter’s input: a .txt file with the sequence(s) to be analyzed. The file layout must comply with the following format (example shown in Fig. 2):
* sequence name preceded by “>” sign, line break
* sequence without spaces, double line break.

Imagen que contiene texto

Descripción generada automáticamente

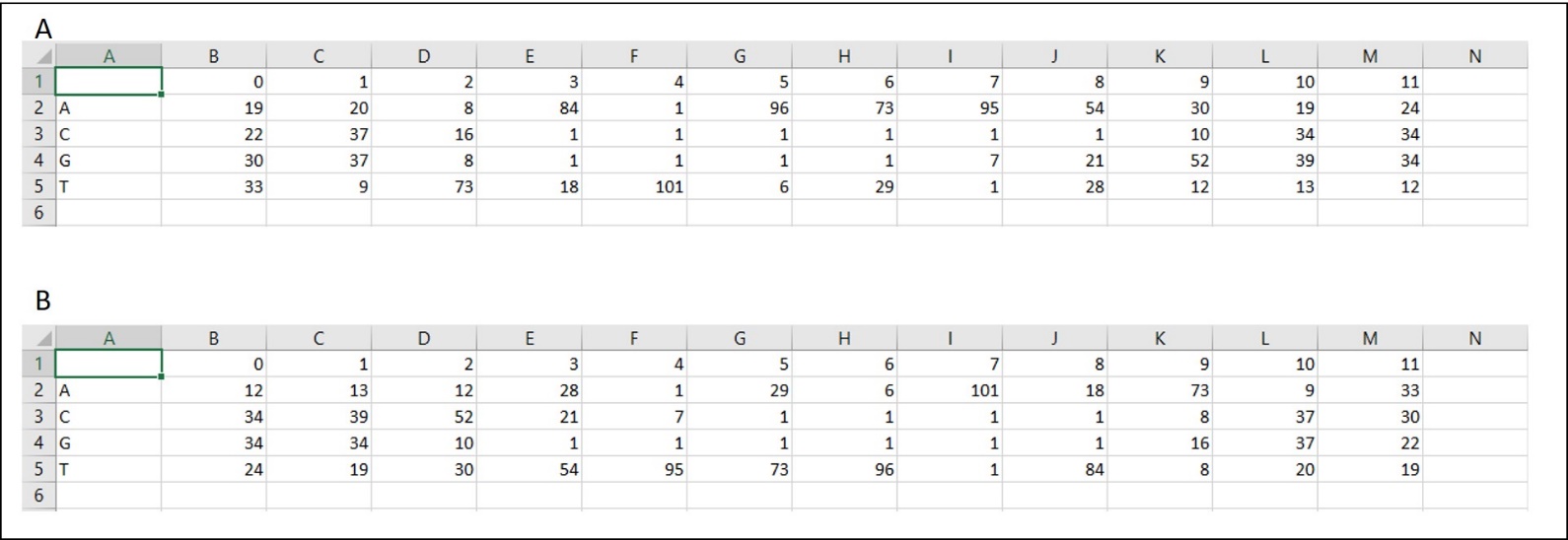
Figure

* A folder named “figuras”, in which Morgan Plotter saves the graphical representation of each analyzed clone
* A .csv file named “tabla\_hits”. This is the file in which the user specifies the location of each satellite repeat along each one of the sequences to be analyzed and is an input for Morgan Plotter. The file layout must comply with the following format (Fig. 3):
* A column named “hit\_id”, where an id is provided to each satellite repeat along the clone
* A column named “clone\_id”, where the name of the clone where each satellite repeat is located
* A column named “identity\_percentage”, where the similarity between each satellite repeat and the consensus sequence of that specific satellite is displayed
* A column named “hit\_start”, which specifies the first base of each satellite repeat along the clone
* A column named “hit\_end”, which specifies the last base of each satellite repeat along the clone



Figure

* A .csv file named “tabla1” which contains the motif to be found, in the 5’-3’ orientation, represented as a Position Specific Frequency Matrix. The table should only contain integers, and the *pseudocount* value is 1 [1]. This is an input for Morgan Plotter and Morgan Score (Fig. 4A).
* A .csv file named “tabla2” which contains the motif to be found, in the 3’-5’ orientation, represented as a Position Specific Frequency Matrix. The table should only contain integers, and the *pseudocount* value is 1 [1]. This is an input for Morgan Plotter and Morgan Score (Fig. 4B).
* The folder .ipynb\_checkpoints, not to be modified by the user.

**MORGAN HUNTER**

Figure

Running Morgan Hunter

Open Anaconda Navigator and launch the jupyter notebook (Fig. 5A). A browser menu will be displayed in the default web navigator (Fig. 5B). Make sure that the *Files* tab is selected (upper left corner), open the folder containing the Morgan Hunter Tool and double click Morgan\_Hunter.ipynb. The Morgan Hunter interface should appear in a new tab of the web browser (Fig. 5C).

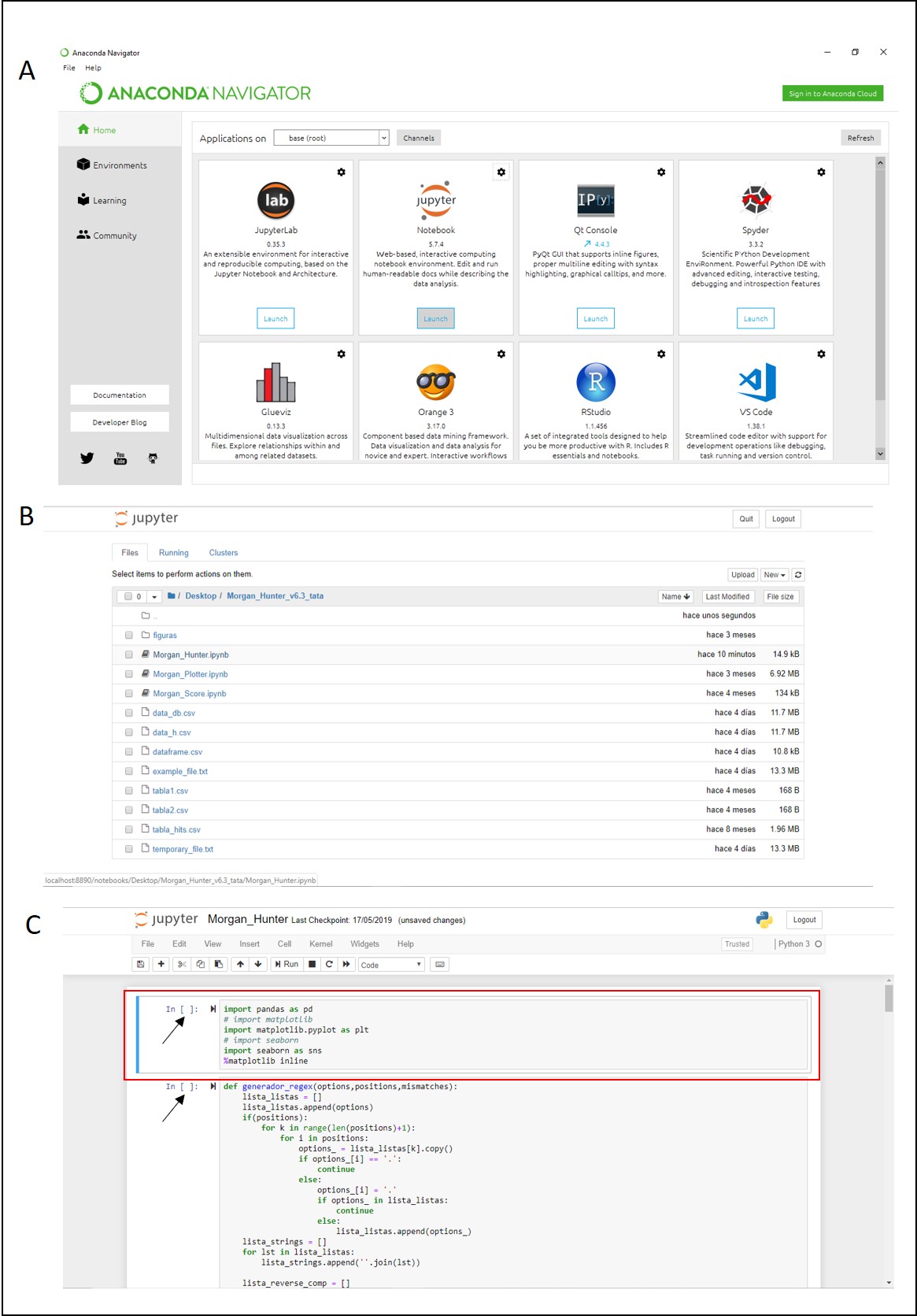
The interface displays a sequence of cells (the first cell is framed by a red rectangle in Fig. 5C) with the commands of the Morgan Hunter algorithm. Cells are run by selecting them (clicking on them) and pressing SHIFT + ENTER. Before a cell is run, brackets to its left appear empty (arrow in Fig. 5C); while the command is running, an asterisk appears inside the brackets; and after the command has been completed, the asterisk is replaced by a number. After a cell is run, the next cell will be automatically selected. If you need to restart running the module, select *Kernel* from the jupyter menu and click on *Restart & Clear Output*.

Before running Morgan Hunter, go to the 10th cell of the algorithm and make sure that the name of the input file (‘example\_file.txt’ in this example, see Fig. 1) is properly written, as shown in the red rectangles in Fig. 6. Go back to the first cell and select it.

Start running the cells of the Morgan Hunter algorithm. The fourth cell requires the user to provide the information of the motif to be found (Fig. 7A, arrow):

* Enter the size of the motif (*Tamaño de la caja*) and press ENTER
* The number of mismatches allowed (*Numero permitido de mismatches*) and press ENTER
* The position of the allowed mismatches separated by commas (*Posiciones para posibles mismatches*) and press ENTER. If no mismatches are allowed, leave the space blank and press ENTER.
* The possible nucleotides of each position along the motif sequence. If two or more nucleotides are allowed in the same position, they should be separated by a pipe (|). Use capital letters only. Press ENTER after each entry (Fig. 7B).

The information provided for the motif search will be displayed on screen (Fig. 7C).

****

Figure

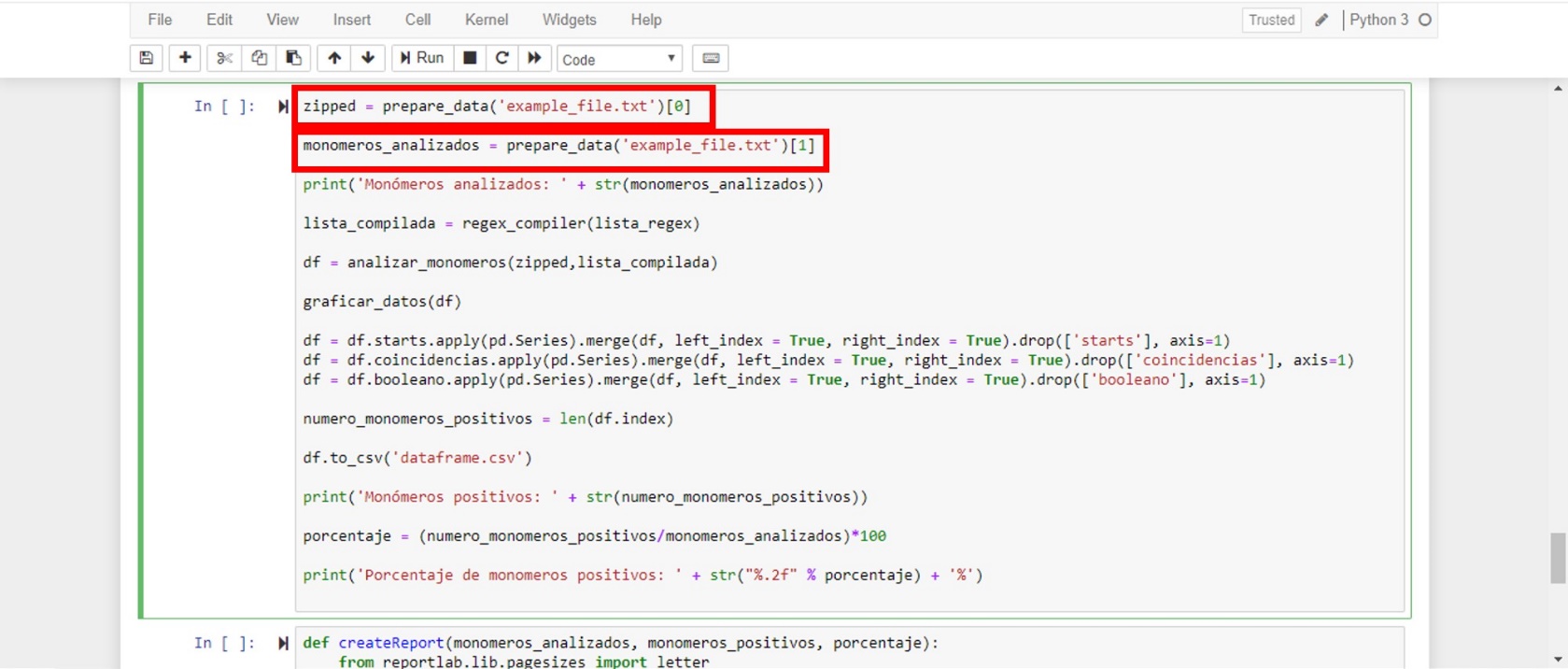


Figure 6

Morgan Hunter results

Morgan Hunter’s outputs are the following:

* The positions (a list of numbers) number (*Monómeros positivos*) and percentage (*Porcentaje de monómeros positivos*) of input sequences that show one or more motifs (displayed on screen below the 10th cell of the algorithm; example shown in Fig. 8)
* A box plot showing the length of the sequences that contain different numbers of motifs (displayed on screen below the 10th cell of the algorithm; example shown in Fig. 8)
* A .csv file named dataframe.csv containing a table showing the coordinates, sequences, and orientation of each motif found along each sequence (saved in the program folder and displayed on screen; example shown in Fig. 9). In this file, each row contains the information of a single sequence:
* The first column assigns a number to the sequence.
* The next n columns (where n = the maximum number of motifs found in a single sequence) indicate whether each of the motifs found is in the 5’-3’ or 3’-5’ orientation (0= 5’-3’ and 1=3’-5’) (Fig. 9A)
* The next n columns show the sequence of each motif found (Fig. 9B)
* The next n columns show the start position of each motif found (Fig. 9C)
* The last 3 columns contain the sequence ID, the number of motifs found in each sequence, and the size of the entire sequence (Fig. 9C)

This dataframe is an input file for Morgan Plotter and Morgan Score.

* Two .csv files named data\_h.csv and data\_db.csv which are required for the following modules of the tool (saved in the program folder; inputs for Morgan Plotter and Morgan Score)**.**

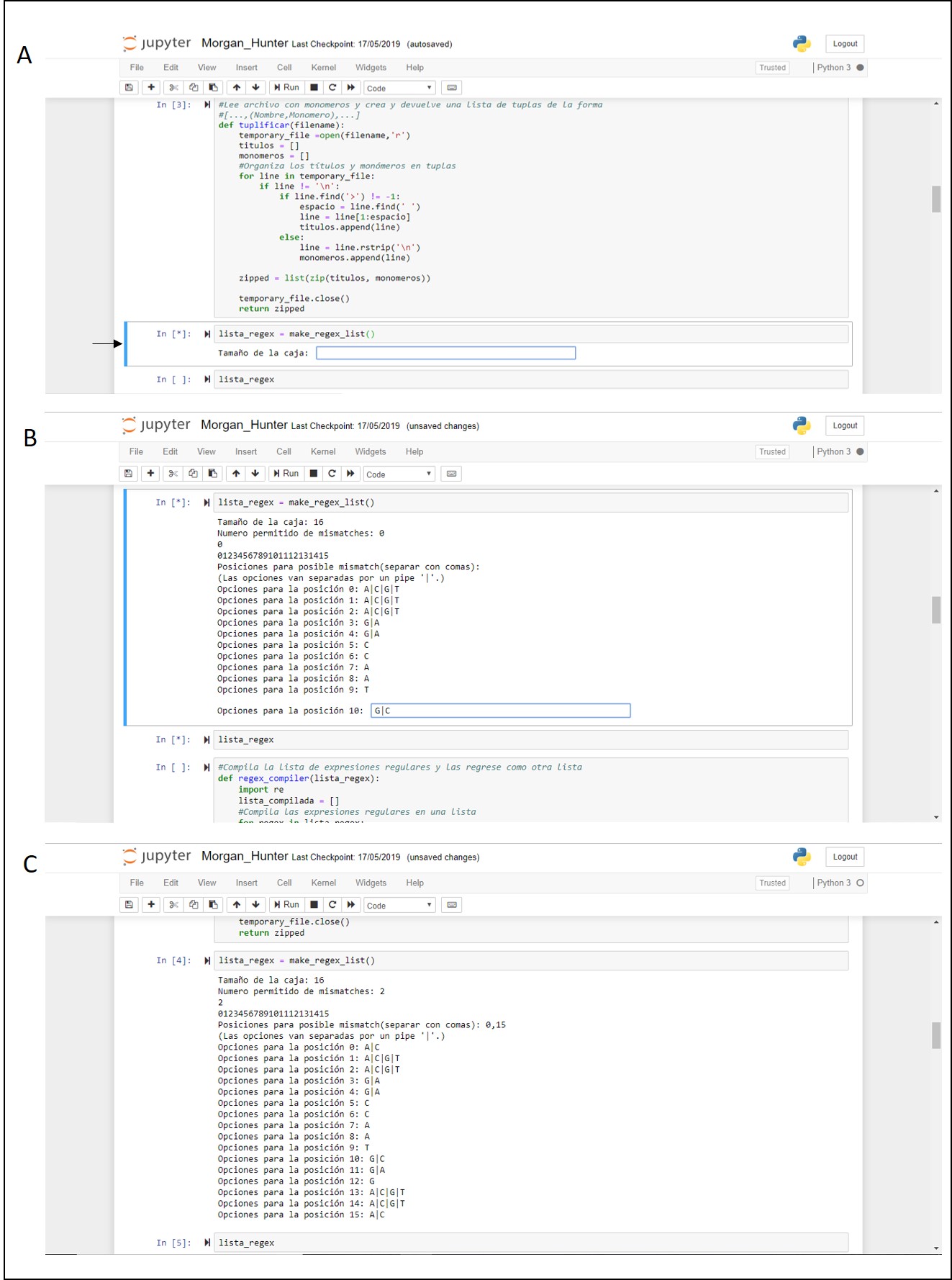


Figure 7

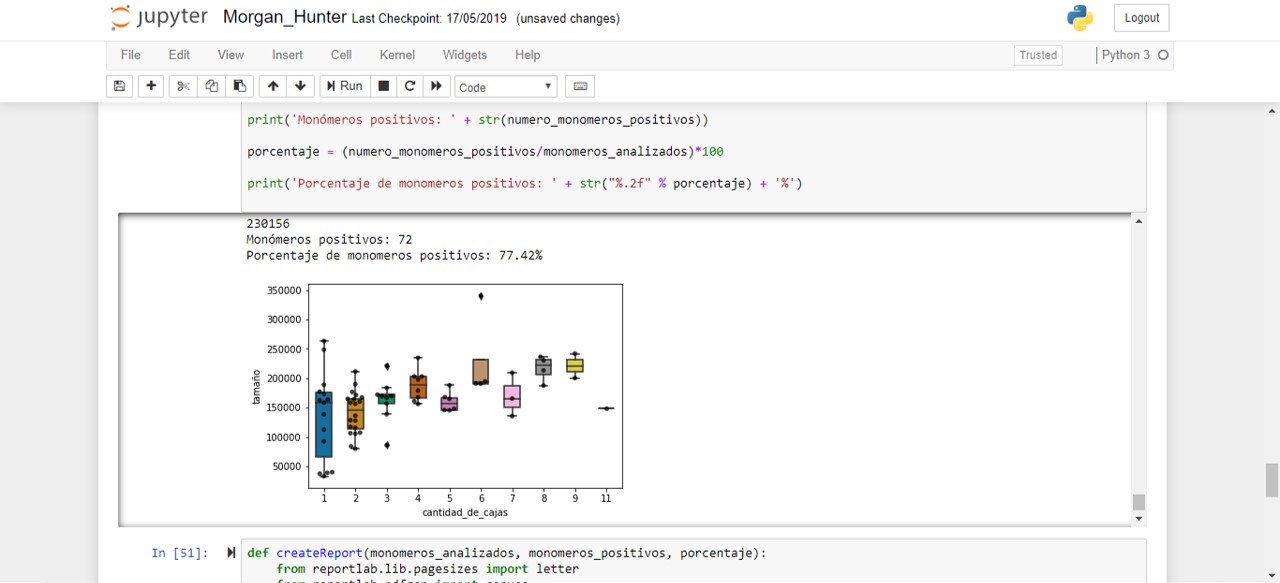
**MORGAN PLOTTER**

Morgan Plotter is opened in the same way as Morgan Hunter. If the Morgan Hunter output files have not been modified, Morgan Plotter can be run by selecting *Cell>Run all* in the jupyter menu or pressing SHIFT + ENTER in each cell, as previously described.

Morgan Plotter results

The numbers that appear under the last cell are the scores of motif/consensus sequence similarity (calculated as in reference [1]) for each motif found (Fig. 10A). These data are also shown in a .csv file named “scores”, saved in the program folder. In this file each row represents an analyzed sequence. The first column contains the name of the sequences, and the following contain the score of each of the motifs detected along the corresponding sequence (Fig. 11).

Below the scores, the resulting graphical representations of each analyzed sequence will be displayed on screen, and a .png image of each one of them will be saved in the folder named “figuras”. For each analyzed sequence, two horizontal grey bars are displayed, the upper one shows the motifs found in the 5’-3’ orientation, while the one below shows the motifs found in the 3’-5’ orientation. Vertical lines that cross each grey bar represent the motifs detected by Morgan Hunter. Their score of similarity to the consensus are color coded: dark green = high similarity; dark red = low similarity. The horizontal black lines delimited by vertical black lines (scale bar symbol) within the grey bars represent the satellite repeats provided by the user (Fig. 10B).



Figure

**Imagen que contiene mapa

Descripción generada automáticamente**

Figure

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figure 10

**MORGAN SCORE**

Imagen que contiene captura de pantalla

Descripción generada automáticamenteMorgan Score is opened and run exactly in the same way as Morgan Plotter. However, it will only generate the consensus similarity scores of each one of the motifs found in the entire input file. This is useful when large amounts of sequences are to be analyzed and no graphical representation is required. With the same input, Morgan Score will run faster than Morgan Plotter. Its output is the same as in Fig. 10A, and it is only displayed on screen.

Figure 11

References

1 Dolfini D, Zambelli F, Pavesi G & Mantovani R (2009) A perspective of promoter architecture from the CCAAT box. *Cell Cycle* 8, 4127–4137.