MORGAN HUNTER TOOL – USER MANUAL

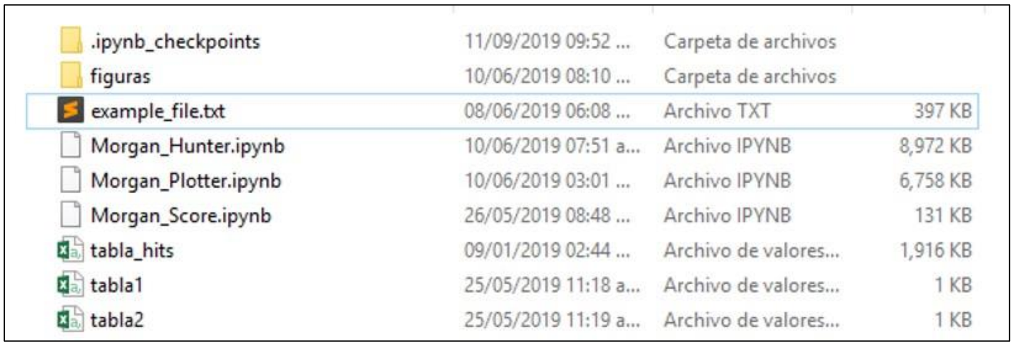
The Morgan Hunter Tool is a Python-based computational tool that searches for motifs on DNA sequences using consensus sequences represented as Position Specific Frequency Matrices. The program provides some of the results in .csv files, others as graphical representations of the analyzed sequence(s),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

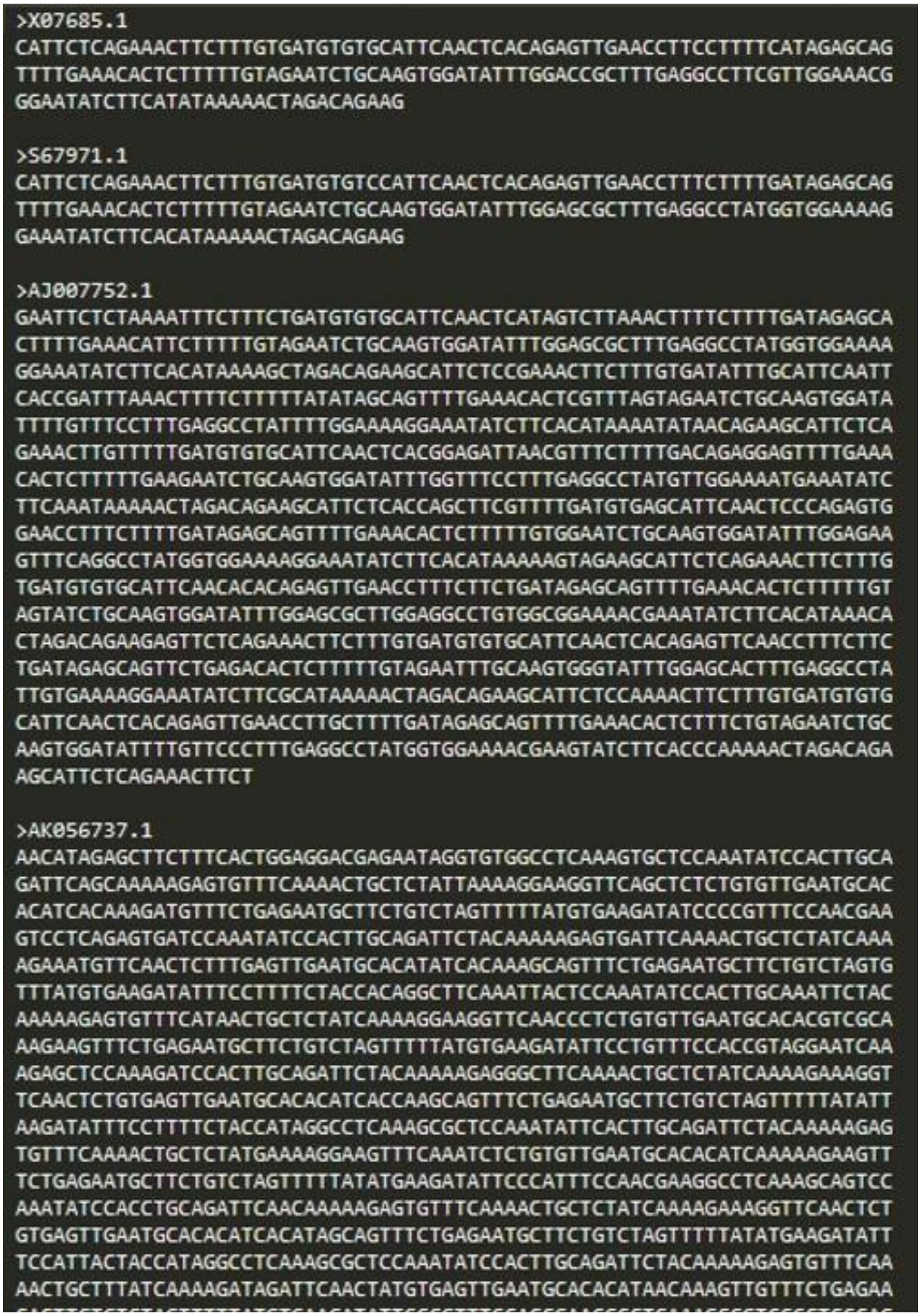
Download Anaconda Distribution (https://www.anaconda.com/distribution/). Make sure that the folder containing the three Morgan Hunter Tool modules (Morgan_Hunter.ipynb, Morgan_Plotter.ipynb, and Morgan_Score.ipynb) also contains the following items (Fig. 1):



.ipynb_checkpoints	11/09/2019 09:52 ...	Carpeta de archivos	
figuras	10/06/2019 08:10 ...	Carpeta de archivos	
example_file.txt	08/06/2019 06:08 ...	Archivo TXT	397 KB
Morgan_Hunter.ipynb	10/06/2019 07:51 a...	Archivo IPYNB	8,972 KB
Morgan_Plotter.ipynb	10/06/2019 03:01 ...	Archivo IPYNB	6,758 KB
Morgan_Score.ipynb	26/05/2019 08:48 ...	Archivo IPYNB	131 KB
tabla_hits	09/01/2019 02:44 ...	Archivo de valores...	1,916 KB
tabla1	25/05/2019 11:18 a...	Archivo de valores...	1 KB
tabla2	25/05/2019 11:19 a...	Archivo de valores...	1 KB

Figure 1

- 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.

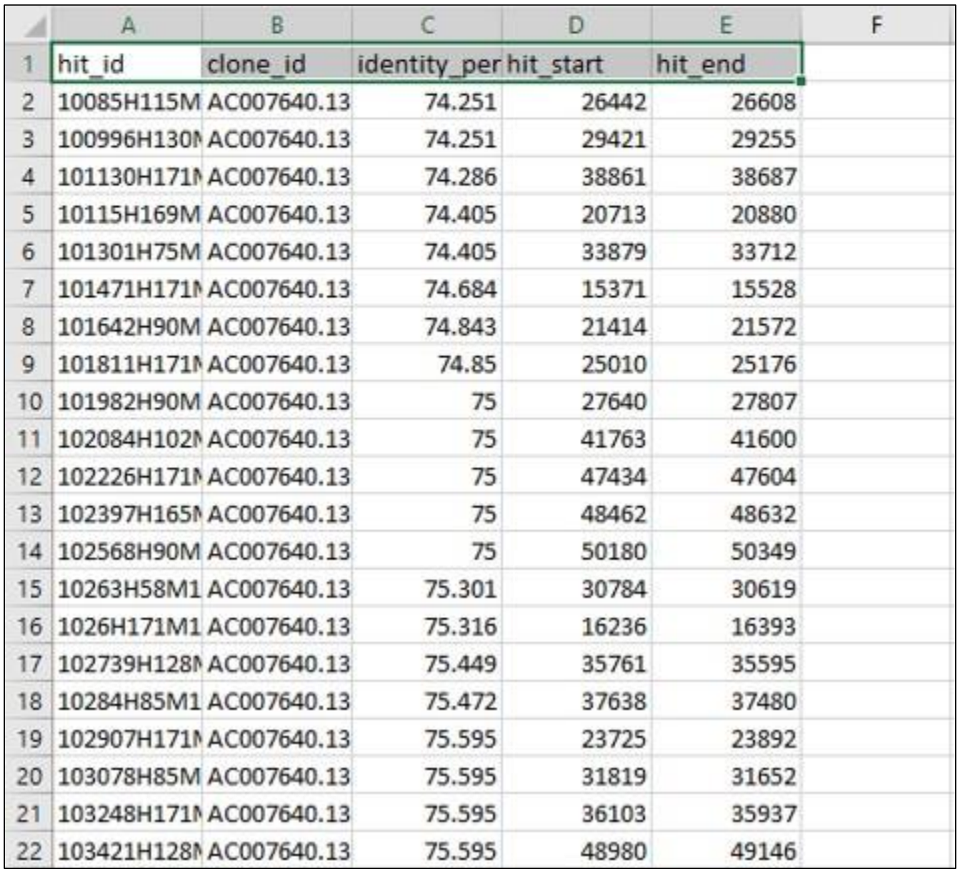


```
>X07685.1
CATTCTCAGAAACTTCTTTGTGATGTGTGCATTCAACTCACAGAGTTGAACCTTCCTTTTCATAGAGCAG
TTTTGAAACACTCTTTTTGTAGAATCTGCAAGTGGATATTTGGACCGCTTTGAGGCCTTCGTTGGAAACG
GGAATATCTTCATATAAAAACTAGACAGAAG

>S67971.1
CATTCTCAGAAACTTCTTTGTGATGTGTCCATTCAACTCACAGAGTTGAACCTTTCTTTTGATAGAGCAG
TTTTGAAACACTCTTTTTGTAGAATCTGCAAGTGGATATTTGGAGCGCTTTGAGGCCTATGGTGGAAAAG
GAAATATCTTCACATAAAAACTAGACAGAAG

>AJ007752.1
GAATTCTCTAAAATTTCTTTCTGATGTGTGCATTCAACTCATAGTCTTAAACTTTTCTTTTGATAGAGCA
CTTTTGAAACATTCTTTTTGTAGAATCTGCAAGTGGATATTTGGAGCGCTTTGAGGCCTATGGTGGAAAA
GGAAATATCTTCACATAAAAGCTAGACAGAAGCATTCTCCGAAACTTCTTTGTGATATTTGCATTCAATT
CACCGATTTAAACTTTTCTTTTTATATAGCAGTTTTGAAACACTCGTTTAGTAGAATCTGCAAGTGGATA
TTTTGTTTCCTTTGAGGCCTATTTTGGAAAAGGAAATATCTTCACATAAAATATAACAGAAGCATTCTCA
GAAACTTGTTTTTGATGTGTGCATTCAACTCACGGAGATTAACGTTTCTTTTGACAGAGGAGTTTTGAAA
CACTCTTTTTGAAGAATCTGCAAGTGGATATTTGGTTTCCTTTGAGGCCTATGTTGGAAAATGAAATATC
TTCAAATAAAAACTAGACAGAAGCATTCTCACCAGCTTCGTTTTGATGTGAGCATTCAACTCCCAGAGTG
GAACCTTTCTTTTGATAGAGCAGTTTTGAAACACTCTTTTTGTGGAATCTGCAAGTGGATATTTGGAGAA
GTTTCAGGCCTATGGTGGAAAAGGAAATATCTTCACATAAAAAGTAGAAGCATTCTCAGAAACTTCTTTG
TGATGTGTGCATTCAACACACAGAGTTGAACCTTTCTTCTGATAGAGCAGTTTTGAAACACTCTTTTTGT
AGTATCTGCAAGTGGATATTTGGAGCGCTTGGAGGCCTGTGGCGGAAAACGAAATATCTTCACATAAACA
CTAGACAGAAGAGTTCTCAGAAACTTCTTTGTGATGTGTGCATTCAACTCACAGAGTTCAACCTTTCTTC
TGATAGAGCAGTTCTGAGACACTCTTTTTGTAGAATTTGCAAGTGGGTATTTGGAGCACTTTGAGGCCTA
TTGTGAAAAGGAAATATCTTCGCATAAAAACTAGACAGAAGCATTCTCCAAAACTTCTTTGTGATGTGTG
CATTCAACTCACAGAGTTGAACCTTGCTTTTGATAGAGCAGTTTTGAAACACTCTTTCTGTAGAATCTGC
AAGTGGATATTTTGTTCCCTTTGAGGCCTATGGTGGAAAACGAAGTATCTTCACCCAAAAACTAGACAGA
AGCATTCTCAGAAACTTCT

>AK056737.1
AACATAGAGCTTCTTTCACTGGAGGACGAGAATAGGTGTGGCCTCAAAGTGCTCCAAATATCCACTTGCA
GATTCAGCAAAAAGAGTGTTTCAAAACTGCTCTATTAAAAGGAAGGTTCAGCTCTCTGTGTTGAATGCAC
ACATCACAAAGATGTTTCTGAGAATGCTTCTGTCTAGTTTTTATGTGAAGATATCCCCGTTTCCAACGAA
GTCCTCAGAGTGATCCAAATATCCACTTGCAGATTCTACAAAAAGAGTGATTCAAAACTGCTCTATCAAA
AGAAATGTTCAACTCTTTGAGTTGAATGCACATATCACAAAGCAGTTTCTGAGAATGCTTCTGTCTAGTG
TTTATGTGAAGATATTTCCTTTTCTACCACAGGCTTCAAATTACTCCAAATATCCACTTGCAAATTCTAC
AAAAAGAGTGTTTCATAACTGCTCTATCAAAAGGAAGGTTCAACCCTCTGTGTTGAATGCACACGTCGCA
AAGAAGTTTCTGAGAATGCTTCTGTCTAGTTTTTATGTGAAGATATTCCTGTTTCCACCGTAGGAATCAA
AGAGCTCCAAAGATCCACTTGCAGATTCTACAAAAAGAGGGCTTCAAAACTGCTCTATCAAAAGAAAGGT
TCAACTCTGTGAGTTGAATGCACACATCACCAAGCAGTTTCTGAGAATGCTTCTGTCTAGTTTTTATATT
AAGATATTTCCTTTTCTACCATAGGCCTCAAAGCGCTCCAAATATTCACTTGCAGATTCTACAAAAAGAG
TGTTTCAAAACTGCTCTATGAAAAGGAAGTTTCAAATCTCTGTGTTGAATGCACACATCAAAAAGAAGTT
TCTGAGAATGCTTCTGTCTAGTTTTTATATGAAGATATTCCCATTTCCAACGAAGGCCTCAAAGCAGTCC
AAATATCCACCTGCAGATTCAACAAAAAGAGTGTTTCAAAACTGCTCTATCAAAAGAAAGGTTCAACTCT
GTGAGTTGAATGCACACATCACATAGCAGTTTCTGAGAATGCTTCTGTCTAGTTTTTATATGAAGATATT
TCCATTACTACCATAGGCCTCAAAGCGCTCCAAATATCCACTTGCAGATTCTACAAAAAGAGTGTTTCAA
AACTGCTTTATCAAAAGATAGATTCAACTATGTGAGTTGAATGCACACATAACAAAGTTGTTTCTGAGAA
```

Figure 2

- 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):

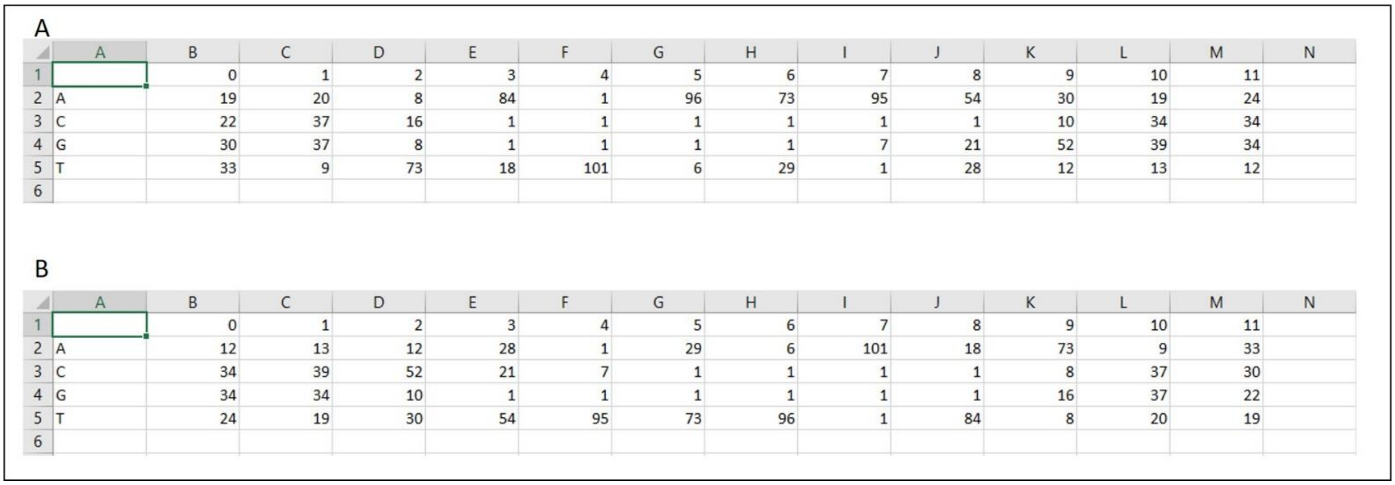
- A column named “hit_id”, where an id is provided to each satellite repeat along the clone
- A column named “clone_id”, where the name of the clone where each satellite repeat is located
- A column named “identity_percentage”, where the similarity between each satellite repeat and the consensus sequence 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



	A	B	C	D	E	F
1	hit_id	clone_id	identity_per	hit_start	hit_end	
2	10085H115M	AC007640.13	74.251	26442	26608	
3	100996H130M	AC007640.13	74.251	29421	29255	
4	101130H171M	AC007640.13	74.286	38861	38687	
5	10115H169M	AC007640.13	74.405	20713	20880	
6	101301H75M	AC007640.13	74.405	33879	33712	
7	101471H171M	AC007640.13	74.684	15371	15528	
8	101642H90M	AC007640.13	74.843	21414	21572	
9	101811H171M	AC007640.13	74.85	25010	25176	
10	101982H90M	AC007640.13	75	27640	27807	
11	102084H102M	AC007640.13	75	41763	41600	
12	102226H171M	AC007640.13	75	47434	47604	
13	102397H165M	AC007640.13	75	48462	48632	
14	102568H90M	AC007640.13	75	50180	50349	
15	10263H58M1	AC007640.13	75.301	30784	30619	
16	1026H171M1	AC007640.13	75.316	16236	16393	
17	102739H128M	AC007640.13	75.449	35761	35595	
18	10284H85M1	AC007640.13	75.472	37638	37480	
19	102907H171M	AC007640.13	75.595	23725	23892	
20	103078H85M	AC007640.13	75.595	31819	31652	
21	103248H171M	AC007640.13	75.595	36103	35937	
22	103421H128M	AC007640.13	75.595	48980	49146	

Figure 3

- 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.



A

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		0	1	2	3	4	5	6	7	8	9	10	11	
2	A	19	20	8	84	1	96	73	95	54	30	19	24	
3	C	22	37	16	1	1	1	1	1	1	10	34	34	
4	G	30	37	8	1	1	1	1	7	21	52	39	34	
5	T	33	9	73	18	101	6	29	1	28	12	13	12	
6														

B

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		0	1	2	3	4	5	6	7	8	9	10	11	
2	A	12	13	12	28	1	29	6	101	18	73	9	33	
3	C	34	39	52	21	7	1	1	1	1	8	37	30	
4	G	34	34	10	1	1	1	1	1	1	16	37	22	
5	T	24	19	30	54	95	73	96	1	84	8	20	19	
6														

Figure 4

MORGAN HUNTER

Running Morgan Hunter

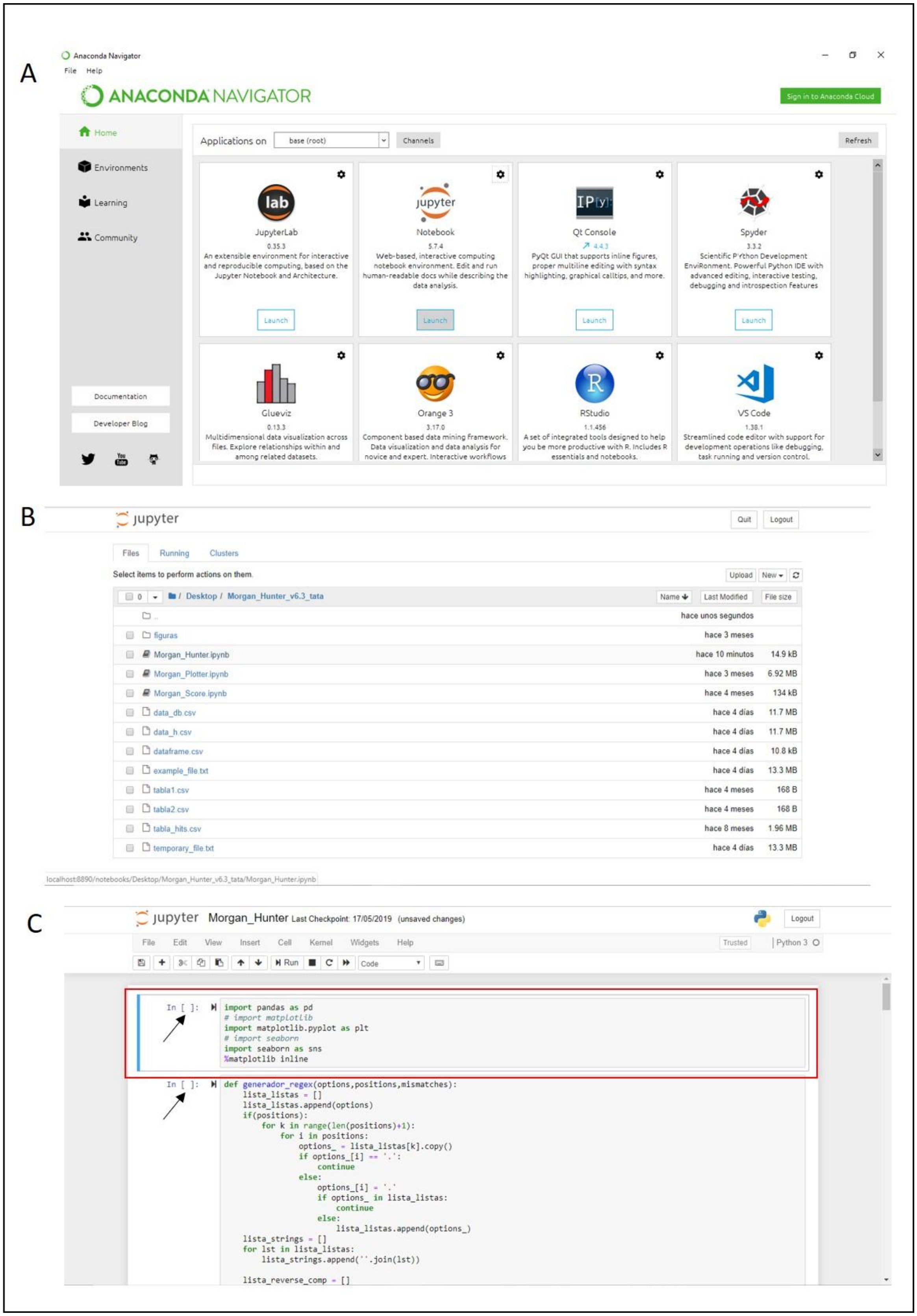
Open Anaconda Navigator and launch the jupyter notebook (Fig. 5A). A browser menu will be displayed in the default web navigator (Fig. 5B). Make sure that the *Files* tab is selected (upper left corner), open the folder containing the Morgan Hunter Tool and double click Morgan_Hunter.ipynb. The Morgan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



JupyterLab
0.35.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
5.7.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console
4.4.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
3.3.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Glueviz
0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3
3.17.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

VS Code
1.38.1
Streamlined code editor with support for development operations like debugging, task running and version control.

Select items to perform actions on them.

/ Desktop / Morgan_Hunter_v6.3_tata

Name	Last Modified	File size
..	hace unos segundos	
figuras	hace 3 meses	
Morgan_Hunter.ipynb	hace 10 minutos	14.9 kB
Morgan_Plotter.ipynb	hace 3 meses	6.92 MB
Morgan_Score.ipynb	hace 4 meses	134 kB
data_db.csv	hace 4 días	11.7 MB
data_h.csv	hace 4 días	11.7 MB
dataframe.csv	hace 4 días	10.8 kB
example_file.txt	hace 4 días	13.3 MB
tabla1.csv	hace 4 meses	168 B
tabla2.csv	hace 4 meses	168 B
tabla_hits.csv	hace 8 meses	1.96 MB
temporary_file.txt	hace 4 días	13.3 MB

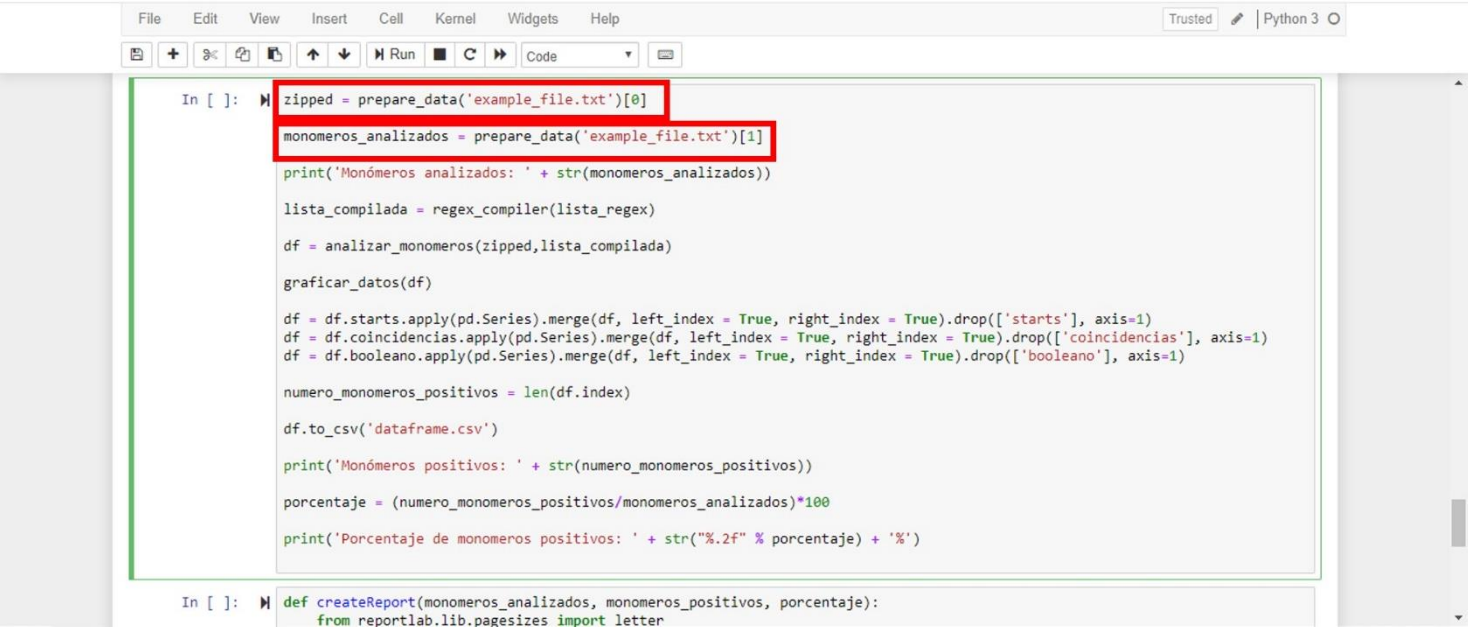
localhost:8890/notebooks/Desktop/Morgan_Hunter_v6.3_tata/Morgan_Hunter.ipynb

jupyter Morgan_Hunter Last Checkpoint: 17/05/2019 (unsaved changes)

```
In [ ]:  import pandas as pd
         # import matplotlib
         import matplotlib.pyplot as plt
         # import seaborn
         import seaborn as sns
         %matplotlib inline
```

```
In [ ]:  def generador_regex(options,positions,mismatches):
             lista_listas = []
             lista_listas.append(options)
             if(positions):
                 for k in range(len(positions)+1):
                     for i in positions:
                         options_ = lista_listas[k].copy()
                         if options_[i] == '.':
                             continue
                         else:
                             options_[i] = '.'
                             if options_ in lista_listas:
                                 continue
                             else:
                                 lista_listas.append(options_)
             lista_strings = []
             for lst in lista_listas:
                 lista_strings.append(''.join(lst))

             lista_reverse_comp = []
```

Figure 5



```
In [ ]:  zipped = prepare_data('example_file.txt')[0]

         monomeros_analizados = prepare_data('example_file.txt')[1]

         print('Monómeros analizados: ' + str(monomeros_analizados))

         lista_compilada = regex_compiler(lista_regex)

         df = analizar_monomeros(zipped,lista_compilada)

         graficar_datos(df)

         df = df.starts.apply(pd.Series).merge(df, left_index = True, right_index = True).drop(['starts'], axis=1)
         df = df.coincidencias.apply(pd.Series).merge(df, left_index = True, right_index = True).drop(['coincidencias'], axis=1)
         df = df.booleano.apply(pd.Series).merge(df, left_index = True, right_index = True).drop(['booleano'], axis=1)

         numero_monomeros_positivos = len(df.index)

         df.to_csv('dataframe.csv')

         print('Monómeros positivos: ' + str(numero_monomeros_positivos))

         porcentaje = (numero_monomeros_positivos/monomeros_analizados)*100

         print('Porcentaje de monomeros positivos: ' + str("%.2f" % porcentaje) + '%')

In [ ]:  def createReport(monomeros_analizados, monomeros_positivos, porcentaje):
             from reportlab.lib.pagesizes import letter
```

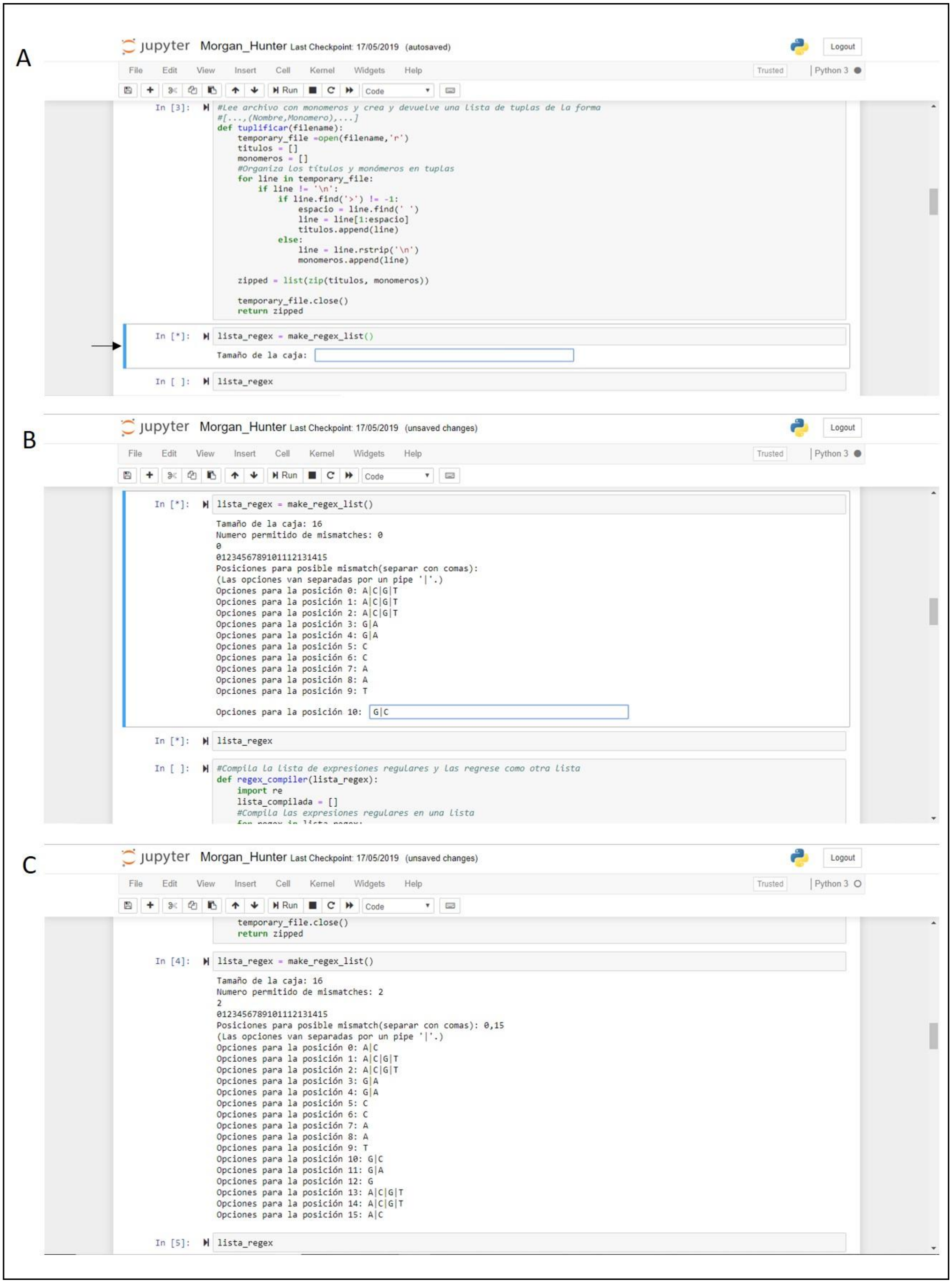
Figure 6

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).


A
jupyter Morgan_Hunter Last Checkpoint: 17/05/2019 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [3]:  #Lee archivo con monomeros y crea y devuelve una lista de tuplas de la forma
         #[...,(Nombre,Monomero),...]
         def tuplificar(filename):
             temporary_file =open(filename,'r')
             titulos = []
             monomeros = []
             #Organiza los títulos y monómeros en tuplas
             for line in temporary_file:
                 if line != '\n':
                     if line.find('>') != -1:
                         espacio = line.find(' ')
                         line = line[1:espacio]
                         titulos.append(line)
                     else:
                         line = line.rstrip('\n')
                         monomeros.append(line)

             zipped = list(zip(titulos, monomeros))

             temporary_file.close()
             return zipped

In [*]:  lista_regex = make_regex_list()

         Tamaño de la caja:

In [ ]:  lista_regex
```

B
jupyter Morgan_Hunter Last Checkpoint: 17/05/2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [*]:  lista_regex = make_regex_list()

         Tamaño de la caja: 16
         Numero permitido de mismatches: 0
         0
         0123456789101112131415
         Posiciones para posible mismatch(separar con comas):
         (Las opciones van separadas por un pipe '|'.)
         Opciones para la posición 0: A|C|G|T
         Opciones para la posición 1: A|C|G|T
         Opciones para la posición 2: A|C|G|T
         Opciones para la posición 3: G|A
         Opciones para la posición 4: G|A
         Opciones para la posición 5: C
         Opciones para la posición 6: C
         Opciones para la posición 7: A
         Opciones para la posición 8: A
         Opciones para la posición 9: T

         Opciones para la posición 10:  G|C

In [*]:  lista_regex

In [ ]:  #Compila la lista de expresiones regulares y las regrese como otra lista
         def regex_compiler(lista_regex):
             import re
             lista_compilada = []
             #Compila las expresiones regulares en una lista
```

C
jupyter Morgan_Hunter Last Checkpoint: 17/05/2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
             temporary_file.close()
             return zipped

In [4]:  lista_regex = make_regex_list()

         Tamaño de la caja: 16
         Numero permitido de mismatches: 2
         2
         0123456789101112131415
         Posiciones para posible mismatch(separar con comas): 0,15
         (Las opciones van separadas por un pipe '|'.)
         Opciones para la posición 0: A|C
         Opciones para la posición 1: A|C|G|T
         Opciones para la posición 2: A|C|G|T
         Opciones para la posición 3: G|A
         Opciones para la posición 4: G|A
         Opciones para la posición 5: C
         Opciones para la posición 6: C
         Opciones para la posición 7: A
         Opciones para la posición 8: A
         Opciones para la posición 9: T
         Opciones para la posición 10: G|C
         Opciones para la posición 11: G|A
         Opciones para la posición 12: G
         Opciones para la posición 13: A|C|G|T
         Opciones para la posición 14: A|C|G|T
         Opciones para la posición 15: A|C

In [5]:  lista_regex
```

Figure 7

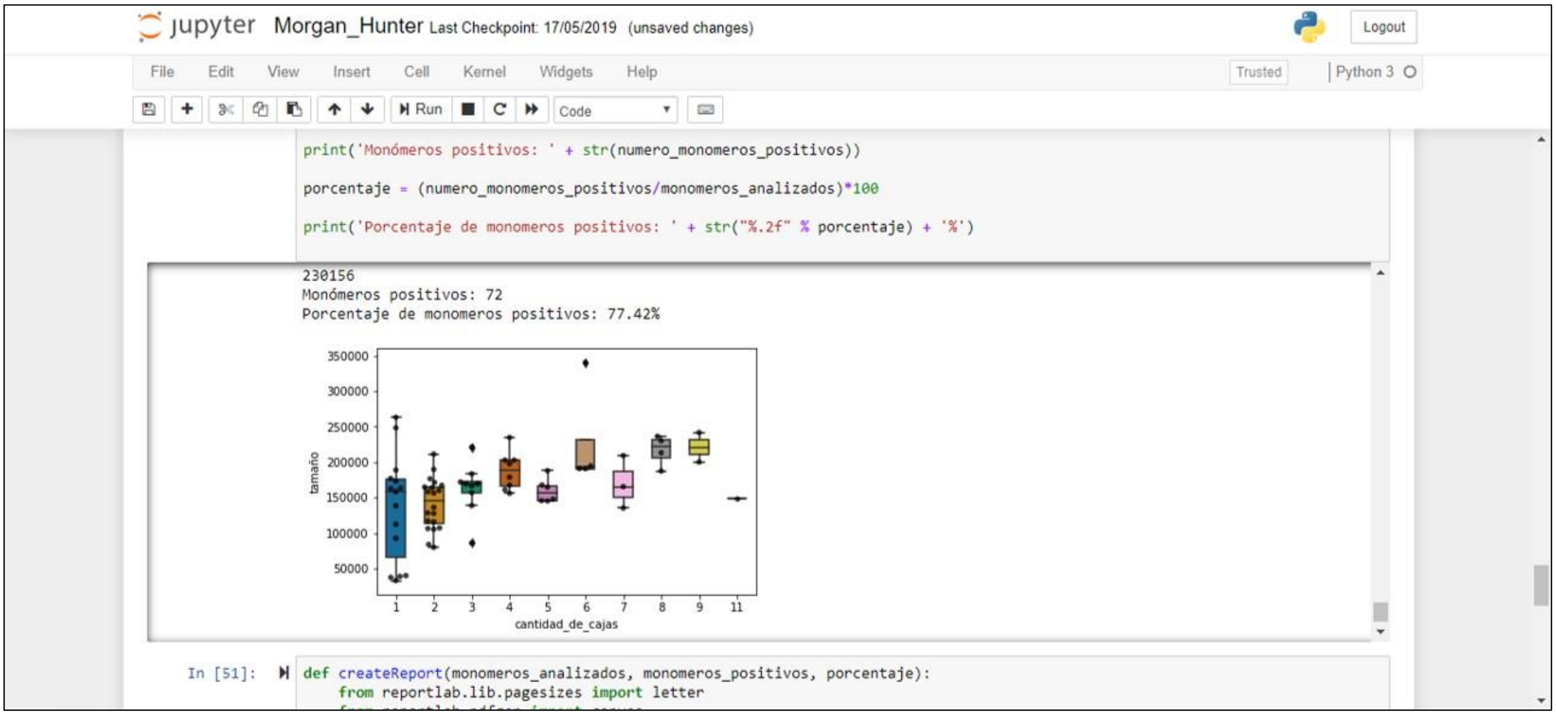
MORGAN PLOTTER

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).

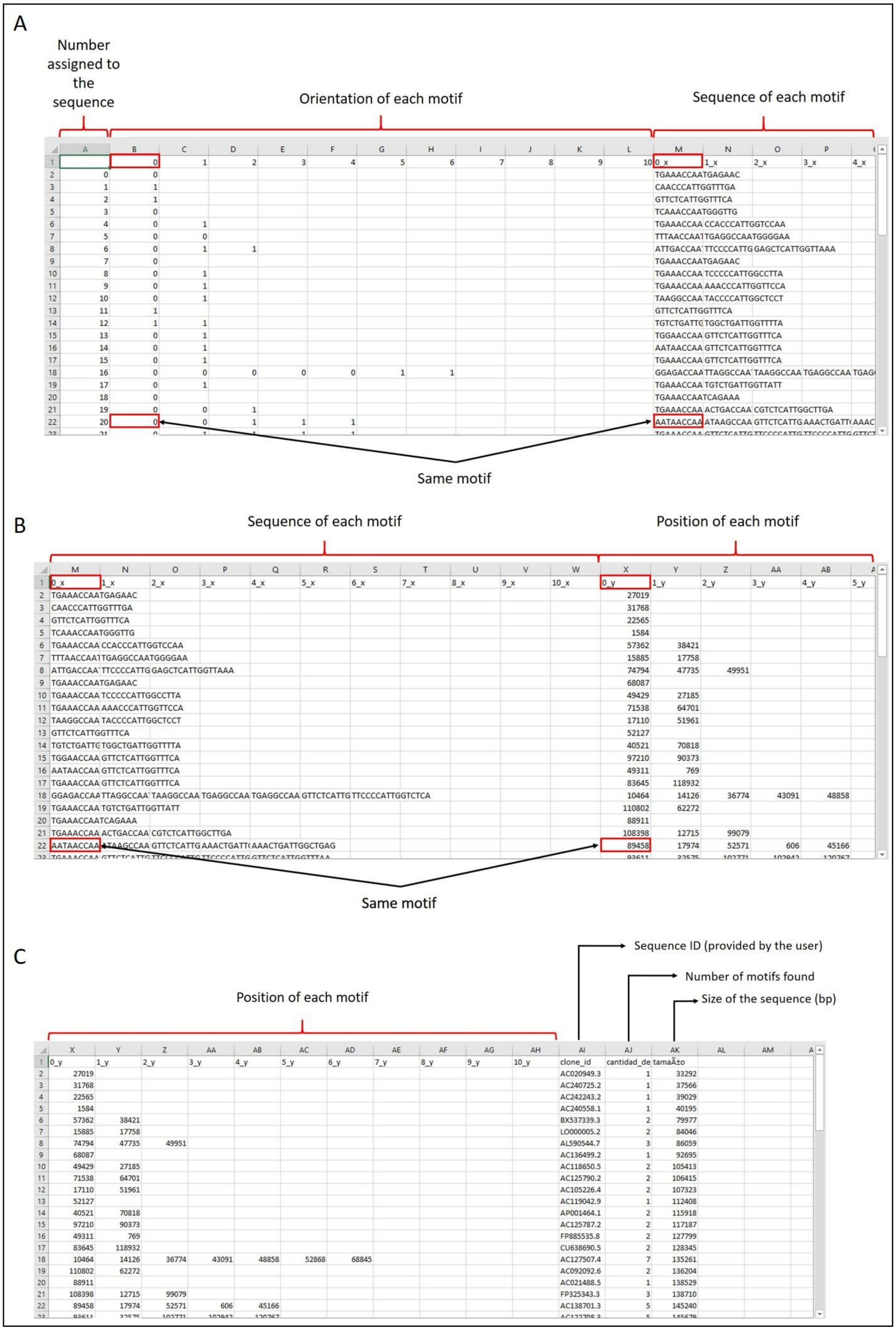


```
print('Monómeros positivos: ' + str(numero_monomeros_positivos))

porcentaje = (numero_monomeros_positivos/monomeros_analizados)*100

print('Porcentaje de monomeros positivos: ' + str("%.2f" % porcentaje) + '%')
```

```
230156
Monómeros positivos: 72
Porcentaje de monomeros positivos: 77.42%
```

```
In [51]:  def createReport(monomeros_analizados, monomeros_positivos, porcentaje):
              from reportlab.lib.pagesizes import letter
```

Figure 8

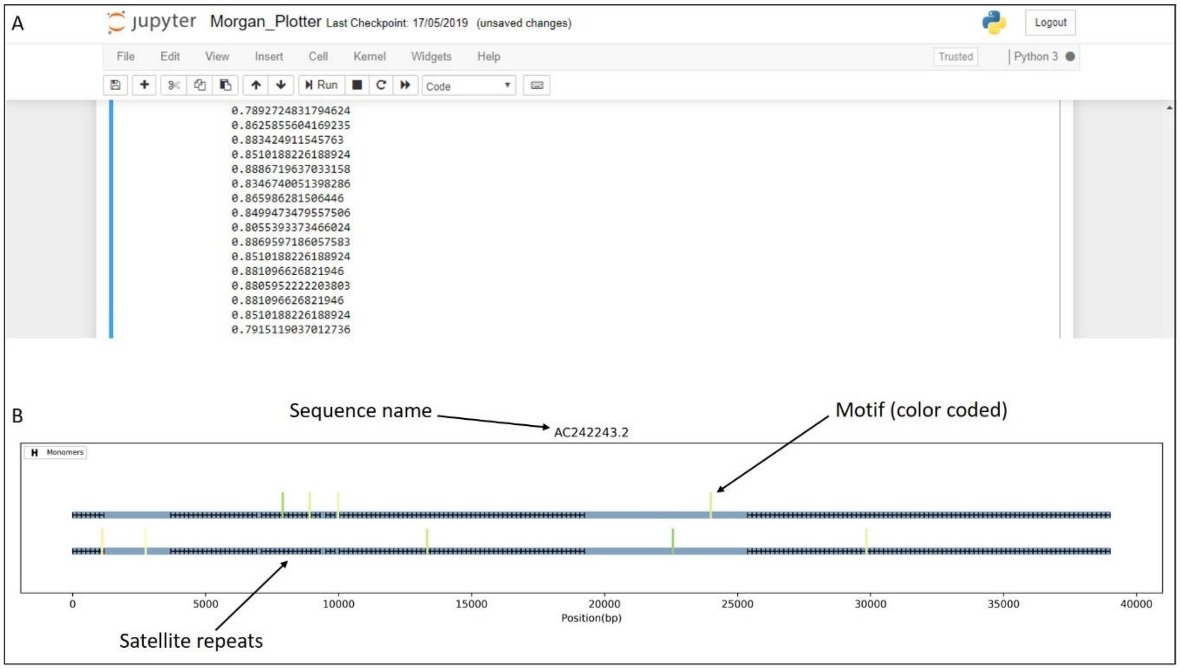


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		0	1	2	3	4	5	6	7	8	9	10	0_x	1_x	2_x	3_x
2	0	0											TGAAACCAATGAGAAC			
3	1	1											CAACCCATTGGTTTGA			
4	2	1											GTTCTCATTGGTTTCA			
5	3	0											TCAAACCAATGGGTTG			
6	4	0	1										TGAAACCAA	CCACCCATTGGTCCAA		
7	5	0	0										TTTAACCAAT	TGAGGCCAATGGGGAA		
8	6	0	1	1									ATTGACCAAT	TTCCCCATTG	GAGCTCATTGGTTAAA	
9	7	0											TGAAACCAATGAGAAC			
10	8	0	1										TGAAACCAA	TCCCCCATTGGCCTTA		
11	9	0	1										TGAAACCAA	AAACCCATTGGTTCCA		
12	10	0	1										TAAGGCCAA	TACCCCATTGGCTCCT		
13	11	1											GTTCTCATTGGTTTCA			
14	12	1	1										TGTCTGATTG	TGGCTGATTGGTTTTA		
15	13	0	1										TGGAACCAA	GTTCTCATTGGTTTCA		
16	14	0	1										AATAACCAA	GTTCTCATTGGTTTCA		
17	15	0	1										TGAAACCAA	GTTCTCATTGGTTTCA		
18	16	0	0	0	0	0	1	1					GGAGACCAA	TTAGGCCAAT	TAAGGCCAA	TGAGGCCAA
19	17	0	1										TGAAACCAA	TGTCTGATTGGTTATT		
20	18	0											TGAAACCAATCAGAAA			
21	19	0	0	1									TGAAACCAA	ACTGACCAAT	CGTCTCATTGGCTTGA	
22	20	0	0	1	1	1							AATAACCAA	ATAAGCCAA	GTTCTCATTG	AAACTGATT
23	21	0	1	1	1	1							TGAAACCAA	GTTCTCATTG	TTCCCCATTG	TTCCCCATTG

	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	10_x	0_y	1_y	2_y	3_y	4_y
2	TGAAACCAATGAGAAC											27019				
3	CAACCCATTGGTTTGA											31768				
4	GTTCTCATTGGTTTCA											22565				
5	TCAAACCAATGGGTTG											1584				
6	TGAAACCAA	CCACCCATTGGTCCAA										57362	38421			
7	TTTAACCAAT	TGAGGCCAATGGGGAA										15885	17758			
8	ATTGACCAAT	TTCCCCATTG	GAGCTCATTGGTTAAA									74794	47735	49951		
9	TGAAACCAATGAGAAC											68087				
10	TGAAACCAA	TCCCCCATTGGCCTTA										49429	27185			
11	TGAAACCAA	AAACCCATTGGTTCCA										71538	64701			
12	TAAGGCCAA	TACCCCATTGGCTCCT										17110	51961			
13	GTTCTCATTGGTTTCA											52127				
14	TGTCTGATTG	TGGCTGATTGGTTTTA										40521	70818			
15	TGGAACCAA	GTTCTCATTGGTTTCA										97210	90373			
16	AATAACCAA	GTTCTCATTGGTTTCA										49311	769			
17	TGAAACCAA	GTTCTCATTGGTTTCA										83645	118932			
18	GGAGACCAA	TTAGGCCAAT	TAAGGCCAA	TGAGGCCAA	TGAGGCCAA	GTTCTCATTG	TTCCCCATTGGTCTCA					10464	14126	36774	43091	48858
19	TGAAACCAA	TGTCTGATTGGTTATT										110802	62272			
20	TGAAACCAATCAGAAA											88911				
21	TGAAACCAA	ACTGACCAAT	CGTCTCATTGGCTTGA									108398	12715	99079		
22	AATAACCAA	ATAAGCCAA	GTTCTCATTG	AAACTGATT	AAACTGATTGGCTGAG							89458	17974	52571	606	45166
23	TGAAACCAA	GTTCTCATTG	TTCCCCATTG	TTCCCCATTG	GTTCTCATTGGTTTAA							93611	32575	102771	102942	120767

	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM
1	0_y	1_y	2_y	3_y	4_y	5_y	6_y	7_y	8_y	9_y	10_y	clone_id	cantidad_de	tamaÃ±o		
2	27019											AC020949.3	1	33292		
3	31768											AC240725.2	1	37566		
4	22565											AC242243.2	1	39029		
5	1584											AC240558.1	1	40195		
6	57362	38421										BX537339.3	2	79977		
7	15885	17758										LO000005.2	2	84046		
8	74794	47735	49951									AL590544.7	3	86059		
9	68087											AC136499.2	1	92695		
10	49429	27185										AC118650.5	2	105413		
11	71538	64701										AC125790.2	2	106415		
12	17110	51961										AC105226.4	2	107323		
13	52127											AC119042.9	1	112408		
14	40521	70818										AP001464.1	2	115918		
15	97210	90373										AC125787.2	2	117187		
16	49311	769										FP885535.8	2	127799		
17	83645	118932										CU638690.5	2	128345		
18	10464	14126	36774	43091	48858	52868	68845					AC127507.4	7	135261		
19	110802	62272										AC092092.6	2	136204		
20	88911											AC021488.5	1	138529		
21	108398	12715	99079									FP325343.3	3	138710		
22	89458	17974	52571	606	45166							AC138701.3	5	145240		
23	93611	32575	102771	102942	120767							AC122708.3	5	145679		

Figure 9

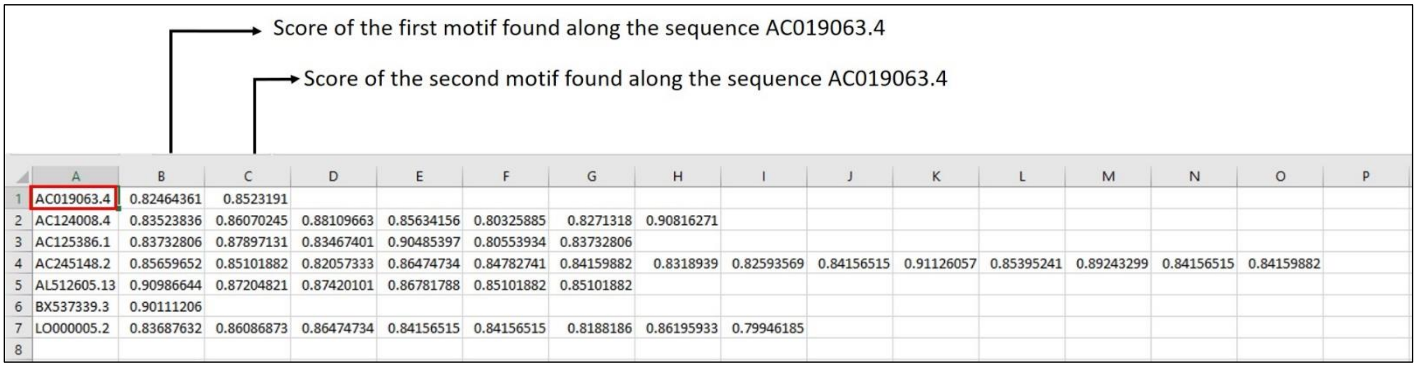


```
0.7892724831794624
0.8625855604169235
0.883424911545763
0.8510188226188924
0.8886719637033158
0.8346740051398286
0.865986281506446
0.8499473479557506
0.8055393373466024
0.8869597186057583
0.8510188226188924
0.881096626821946
0.8805952222203803
0.881096626821946
0.8510188226188924
0.7915119037012736
```

Figure 10

MORGAN SCORE

Morgan 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.



Score of the first motif found along the sequence AC019063.4
Score of the second motif found along the sequence AC019063.4

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	AC019063.4	0.82464361	0.8523191													
2	AC124008.4	0.83523836	0.86070245	0.88109663	0.85634156	0.80325885	0.8271318	0.90816271								
3	AC125386.1	0.83732806	0.87897131	0.83467401	0.90485397	0.80553934	0.83732806									
4	AC245148.2	0.85659652	0.85101882	0.82057333	0.86474734	0.84782741	0.84159882	0.8318939	0.82593569	0.84156515	0.91126057	0.85395241	0.89243299	0.84156515	0.84159882	
5	AL512605.13	0.90986644	0.87204821	0.87420101	0.86781788	0.85101882	0.85101882									
6	BX537339.3	0.90111206														
7	LO000005.2	0.83687632	0.86086873	0.86474734	0.84156515	0.84156515	0.8188186	0.86195933	0.79946185							
8																

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.