PhytoProm:

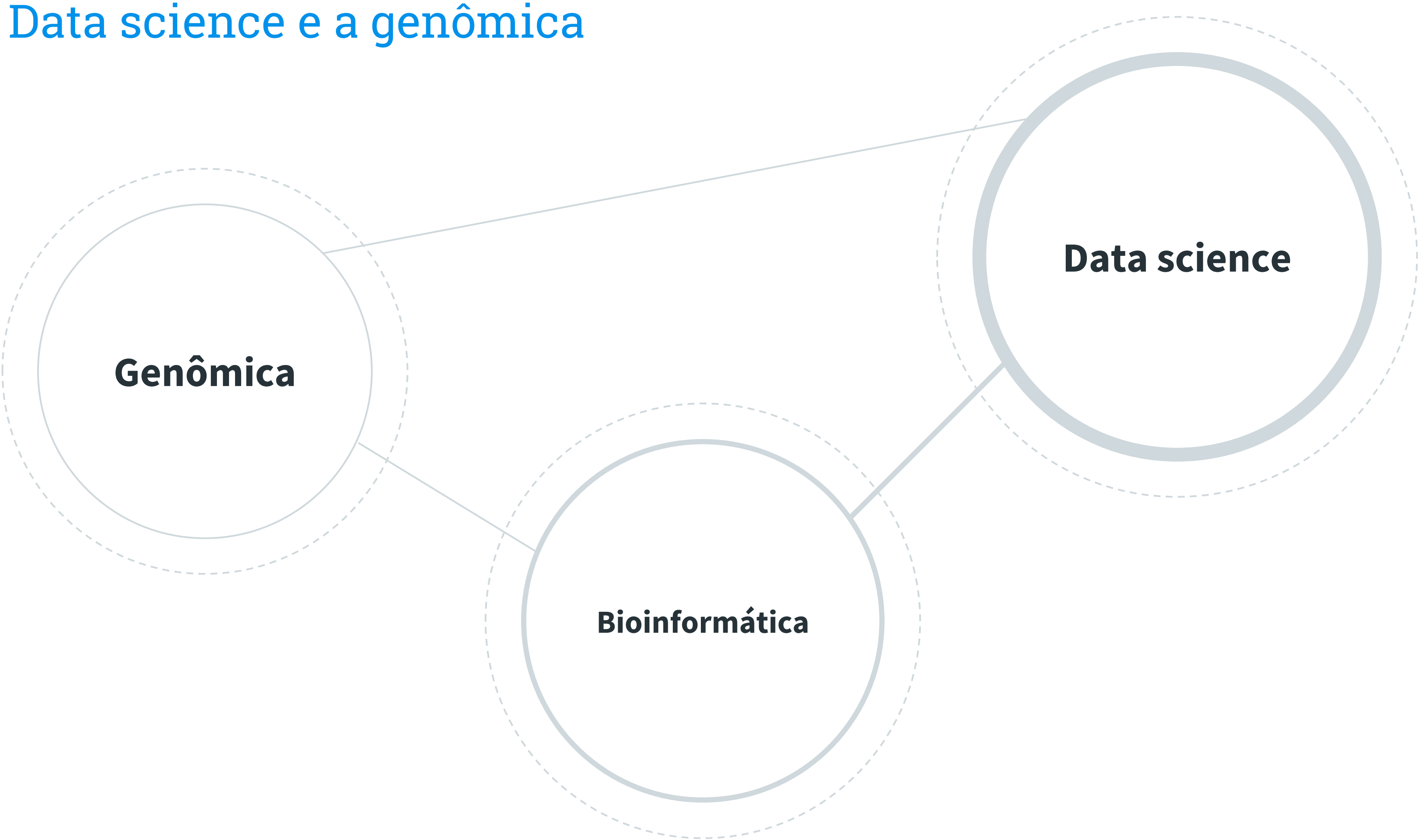
mineração e processamento de DNA para análise de enriquecimento de elementos cis regulatórios

Sobre mim

Filipe Medeiros

- Ciência da computação - UNICAP
- Iniciação científica - Lab. Gen. Biotecnologia Veg. UFPE
- Estagiário - CIn/Motorola

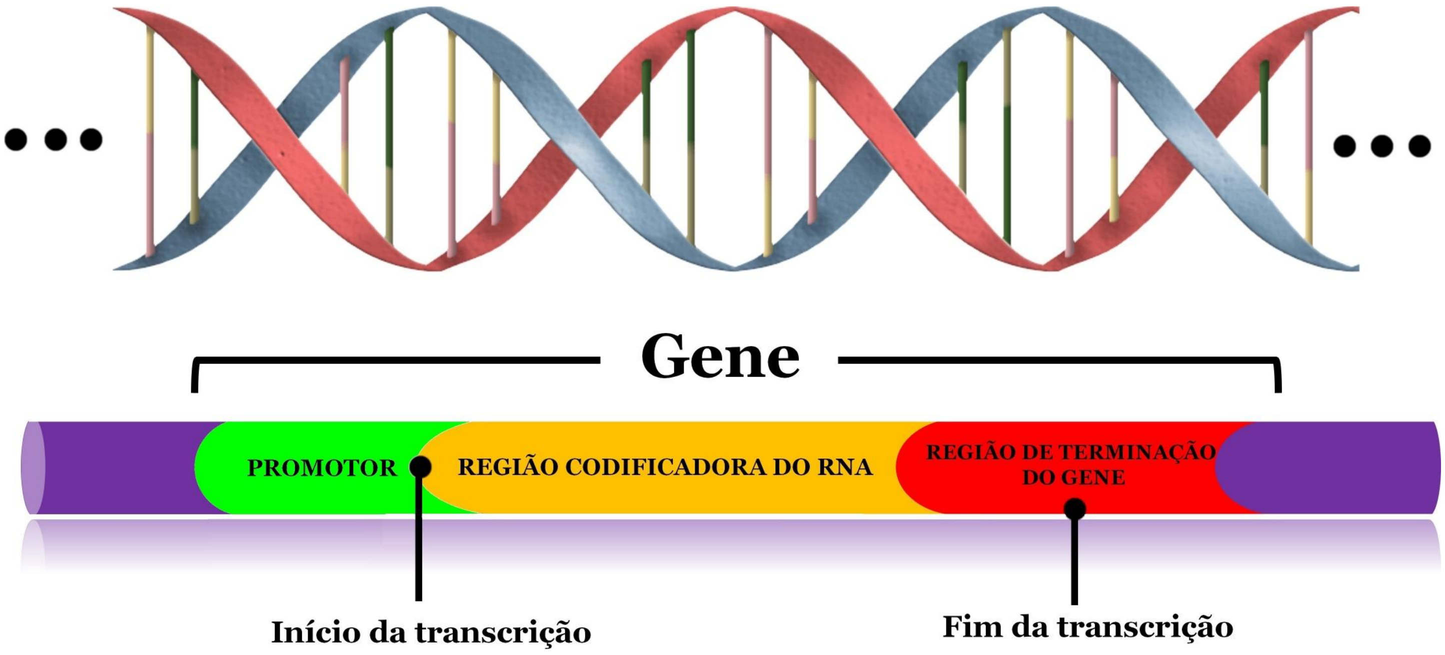
Data science e a genômica



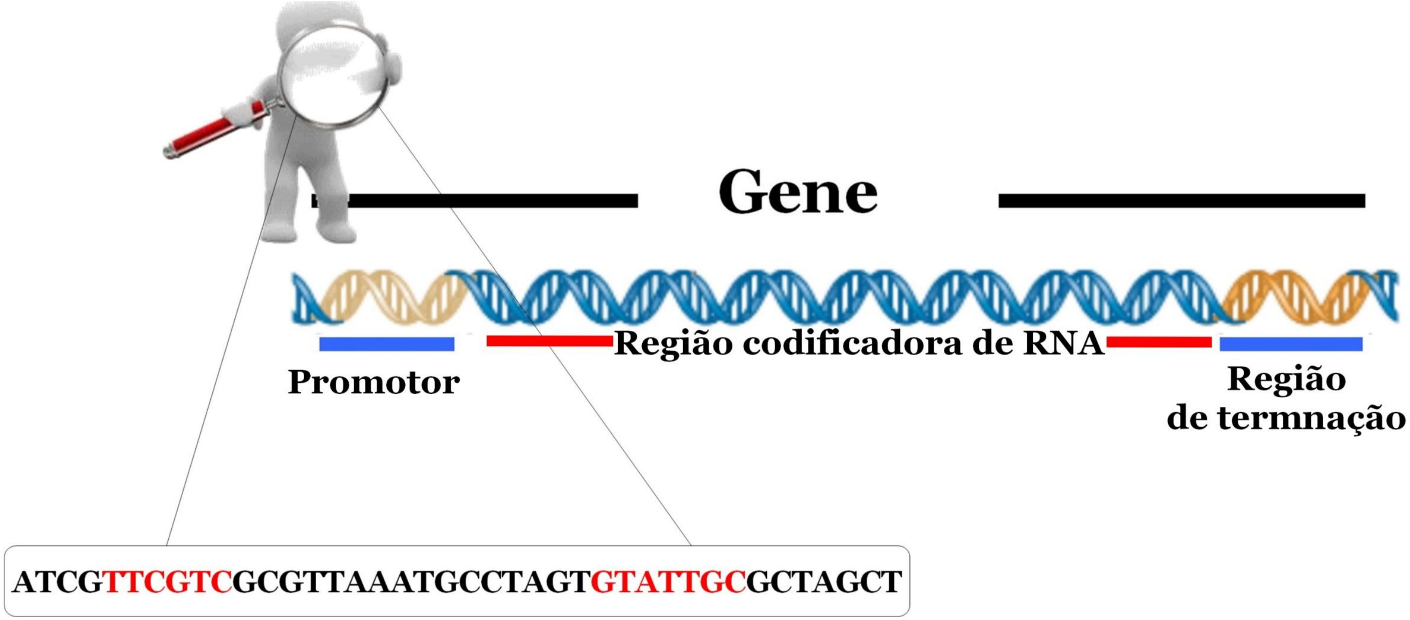
Genoma = conjunto de genes

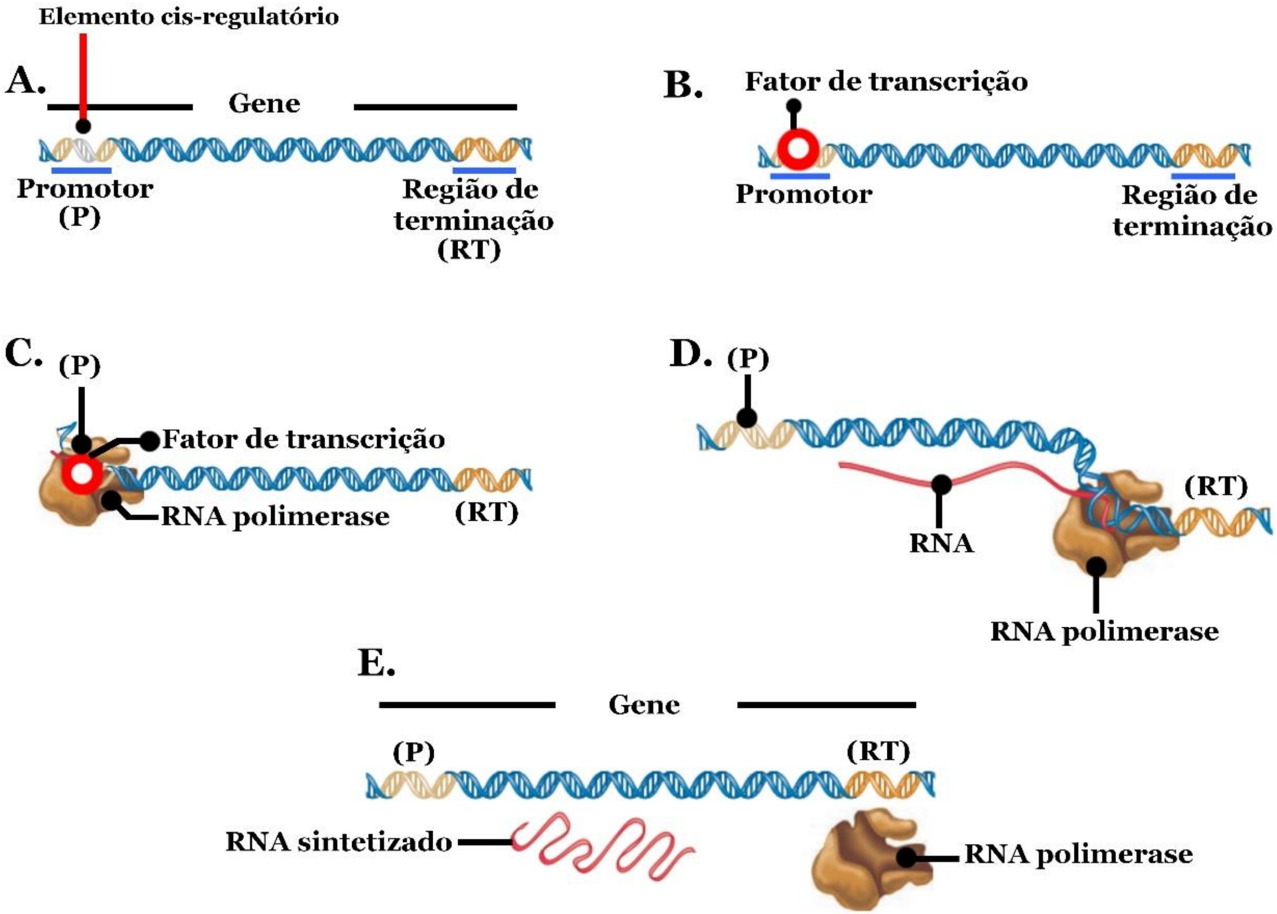
Eficiência energética

Estrutura de um gene



Fator de transcrição


Gene
Região codificadora de RNA
Promotor
Região
de termnação
ATCGTTCGTCGCGTTAAATGCCTAGTGTATTGCGCTAGCT



Processo de síntese de RNA

Mineração dos dados

Jaspar

```
A [13 13 03 01 54 01 01 01 00 03 02 05]
C [13 39 05 53 00 01 50 01 00 37 00 17]
G [17 02 37 00 00 52 03 00 53 08 37 12]
T [11 00 09 00 00 00 00 52 01 06 15 20]
```

Matriz de peso e posição

Tamanho: 489 matrizes

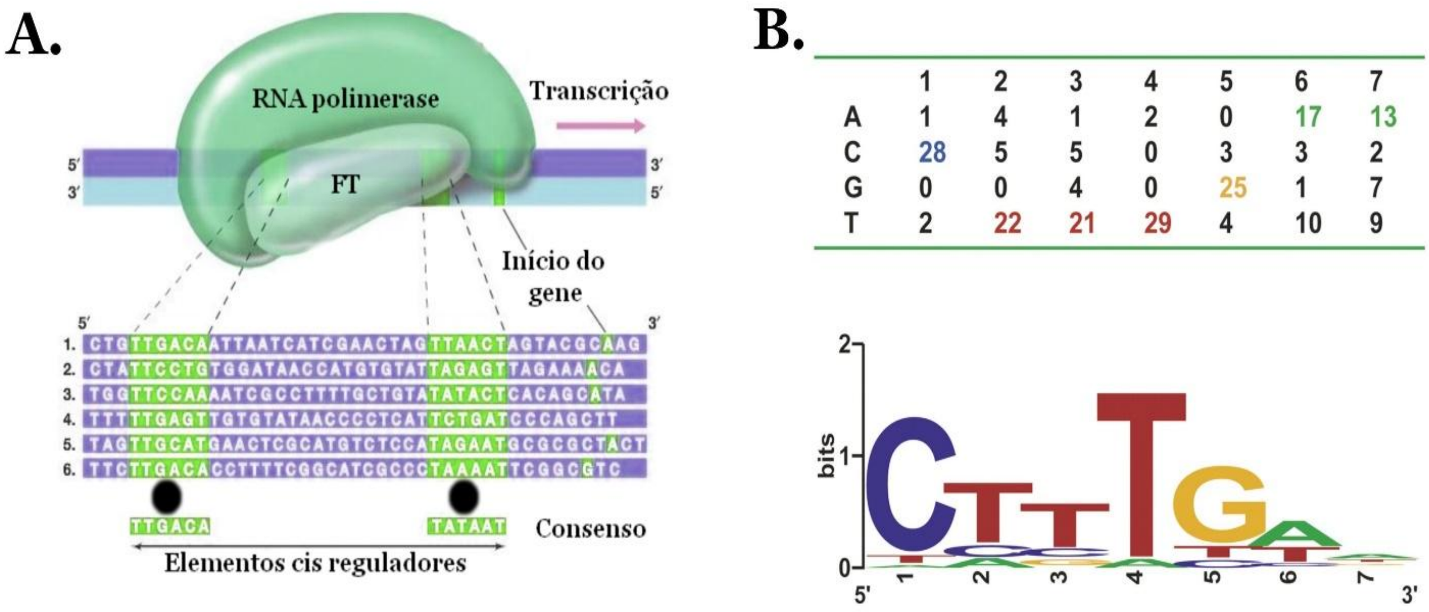
PhytoMine API

```
>Vigun04g127700 5000 upstream
TTTAATCCTTCATCTTTCGAAATACGTG
AATTTAATCATTTTAATCAAATTTTGTTA
AATTTGTTTGATATTTTGTACGTATTTC
ACGATTATATTTGAATTGTTTATAGTGT
TTAACACATTTTTGCTTTAATATTAAGTT
AAATACTATTATAA…
```

FASTA em JSON

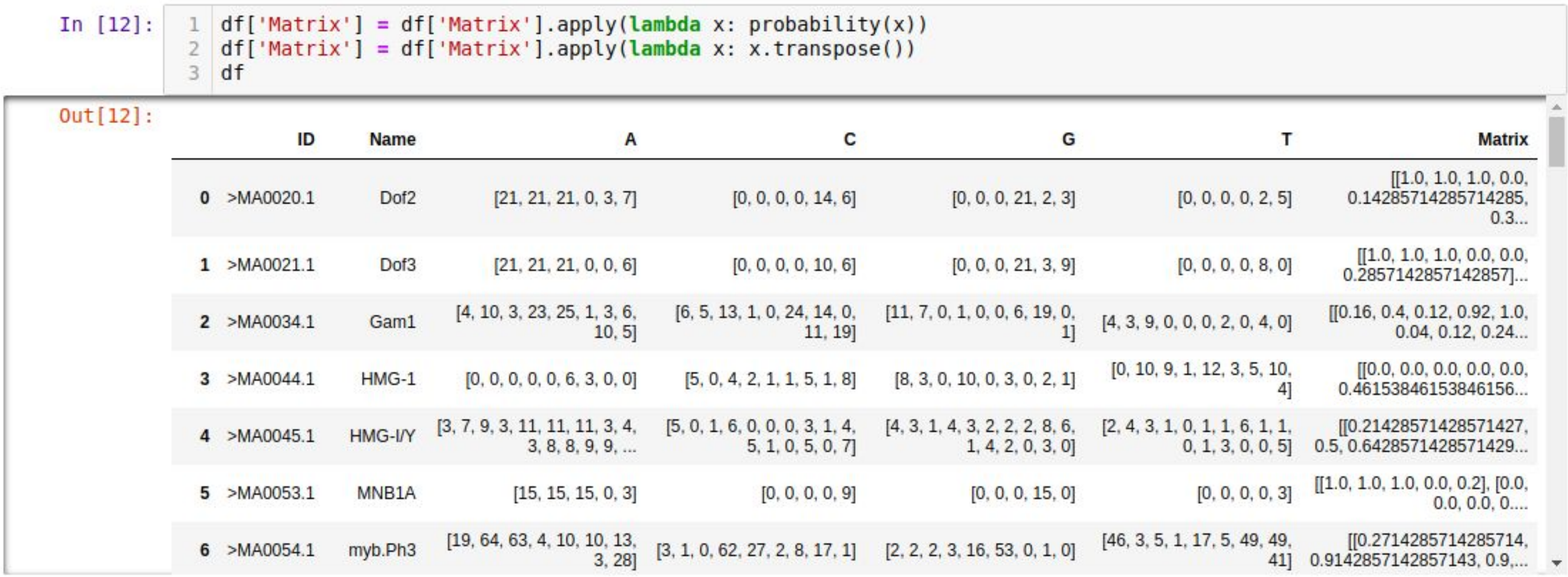
Tamanho: 29.722 promotores

Dados obtidos


A.
RNA polimerase
Transcrição
FT
5′
3′
3′
5′
Início do gene
5′
3′
1. CTGTTGACAATTAATCATCGAACTAGTTAACTAGTACGCAAG
2. CTATTCCTGTGGATAACCATGTGTATTAGAGTTAGAAAACA
3. TGGTTCCAAAATCGCCTTTTGCTGTATATACTCACAGCATA
4. TTTTTGAGTTGTGTATAACCCCTCATTCTGATCCCAGCTT
5. TAGTTGCATGAACTCGCATGTCTCCATAGAATGCGCGCTACT
6. TTCTTGACACCTTTTCGGCATCGCCCTAAAATTCGGCGTC
TTGACA
TATAAT
Consenso
Elementos cis reguladores
B.
2
1
0
bits
1
2
3
4
5
6
7
5'
3'

	1	2	3	4	5	6	7
A	1	4	1	2	0	17	13
C	28	5	5	0	3	3	2
G	0	0	4	0	25	1	7
T	2	22	21	29	4	10	9

Limpeza dos dados

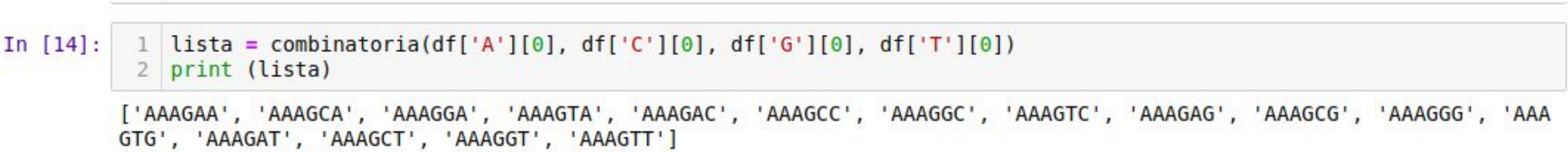


```
In [12]:
df['Matrix'] = df['Matrix'].apply(lambda x: probability(x))
df['Matrix'] = df['Matrix'].apply(lambda x: x.transpose())
df
```

Out[12]:

	ID	Name	A	C	G	T	Matrix
0	>MA0020.1	Dof2	[21, 21, 21, 0, 3, 7]	[0, 0, 0, 0, 14, 6]	[0, 0, 0, 21, 2, 3]	[0, 0, 0, 0, 2, 5]	[[1.0, 1.0, 1.0, 0.0, 0.14285714285714285, 0.3...
1	>MA0021.1	Dof3	[21, 21, 21, 0, 0, 6]	[0, 0, 0, 0, 10, 6]	[0, 0, 0, 21, 3, 9]	[0, 0, 0, 0, 8, 0]	[[1.0, 1.0, 1.0, 0.0, 0.0, 0.2857142857142857]...
2	>MA0034.1	Gam1	[4, 10, 3, 23, 25, 1, 3, 6, 10, 5]	[6, 5, 13, 1, 0, 24, 14, 0, 11, 19]	[11, 7, 0, 1, 0, 0, 6, 19, 0, 1]	[4, 3, 9, 0, 0, 0, 2, 0, 4, 0]	[[0.16, 0.4, 0.12, 0.92, 1.0, 0.04, 0.12, 0.24...
3	>MA0044.1	HMG-1	[0, 0, 0, 0, 0, 6, 3, 0, 0]	[5, 0, 4, 2, 1, 1, 5, 1, 8]	[8, 3, 0, 10, 0, 3, 0, 2, 1]	[0, 10, 9, 1, 12, 3, 5, 10, 4]	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.46153846153846156...
4	>MA0045.1	HMG-I/Y	[3, 7, 9, 3, 11, 11, 11, 3, 4, 3, 8, 8, 9, 9, ...	[5, 0, 1, 6, 0, 0, 0, 3, 1, 4, 5, 1, 0, 5, 0, 7]	[4, 3, 1, 4, 3, 2, 2, 2, 8, 6, 1, 4, 2, 0, 3, 0]	[2, 4, 3, 1, 0, 1, 1, 6, 1, 1, 0, 1, 3, 0, 0, 5]	[[0.21428571428571427, 0.5, 0.6428571428571429...
5	>MA0053.1	MNB1A	[15, 15, 15, 0, 3]	[0, 0, 0, 0, 9]	[0, 0, 0, 15, 0]	[0, 0, 0, 0, 3]	[[1.0, 1.0, 1.0, 0.0, 0.2], [0.0, 0.0, 0.0, 0....
6	>MA0054.1	myb.Ph3	[19, 64, 63, 4, 10, 10, 13, 3, 28]	[3, 1, 0, 62, 27, 2, 8, 17, 1]	[2, 2, 2, 3, 16, 53, 0, 1, 0]	[46, 3, 5, 1, 17, 5, 49, 49, 41]	[[0.2714285714285714, 0.9142857142857143, 0.9,...

Extrair elementos cis regulatórios



```
In [14]: lista = combinatoria(df['A'][0], df['C'][0], df['G'][0], df['T'][0])
         print (lista)
```

```
['AAAGAA', 'AAAGCA', 'AAAGGA', 'AAAGTA', 'AAAGAC', 'AAAGCC', 'AAAGGC', 'AAAGTC', 'AAAGAG', 'AAAGCG', 'AAAGGG', 'AAA
GTG', 'AAAGAT', 'AAAGCT', 'AAAGGT', 'AAAGTT']
```

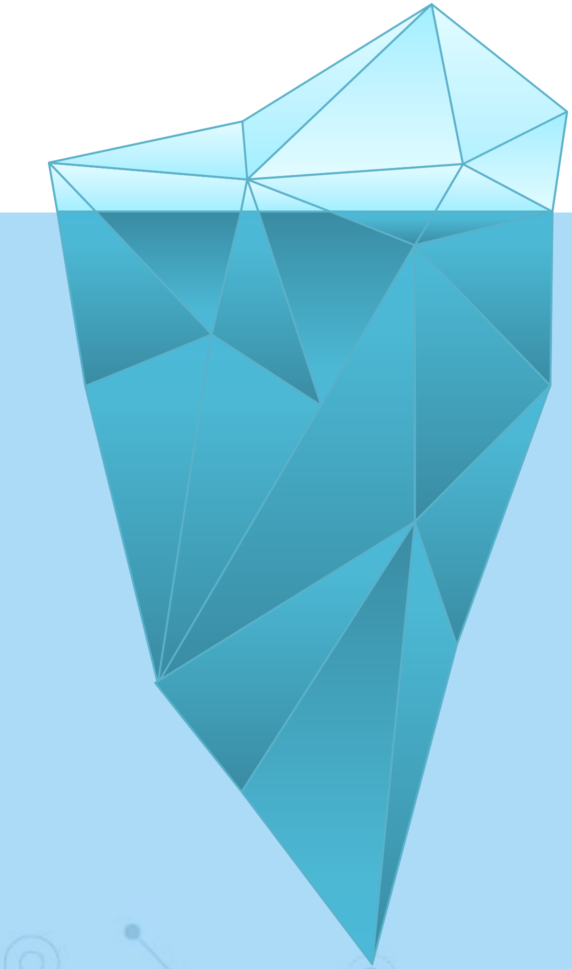
$$4^{30} = 1{,}15.10^{18}$$

Impraticável

Problemas

Grande número de elementos cis regulatórios

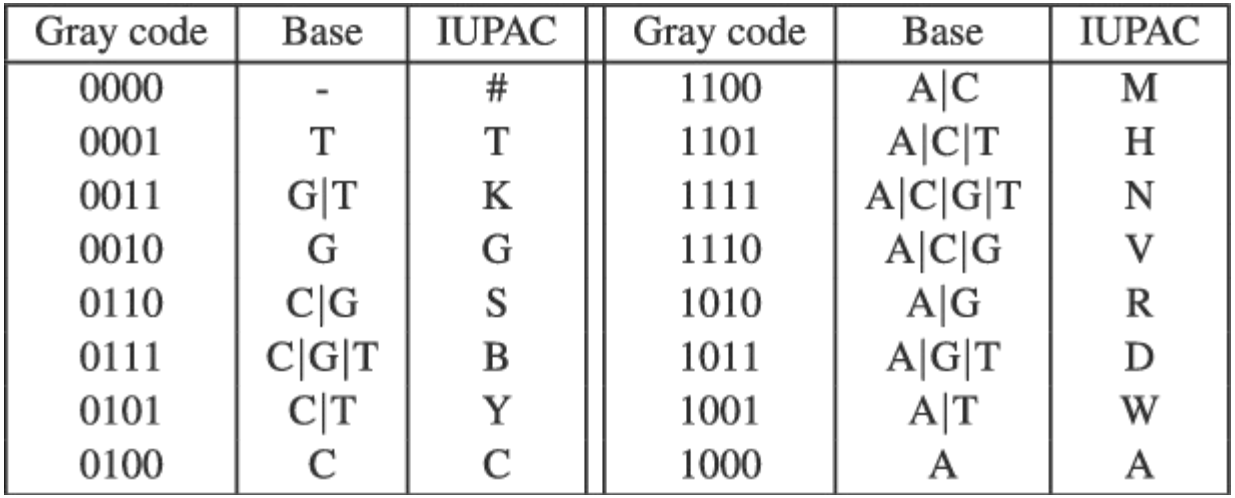
A busca de um grande número de sub-strings torna o processamento do modelo inviável mesmo para um único promotor.



Dupla fita do DNA

Um DNA possui duas fitas gênicas: uma positiva e outra negativa. A análise se concentra apenas na fita positiva para poupar processamento. Isso implica que deve ser feita a busca pelo elemento cis regulatório em forma de complemento reverso para emular a busca na fita negativa.

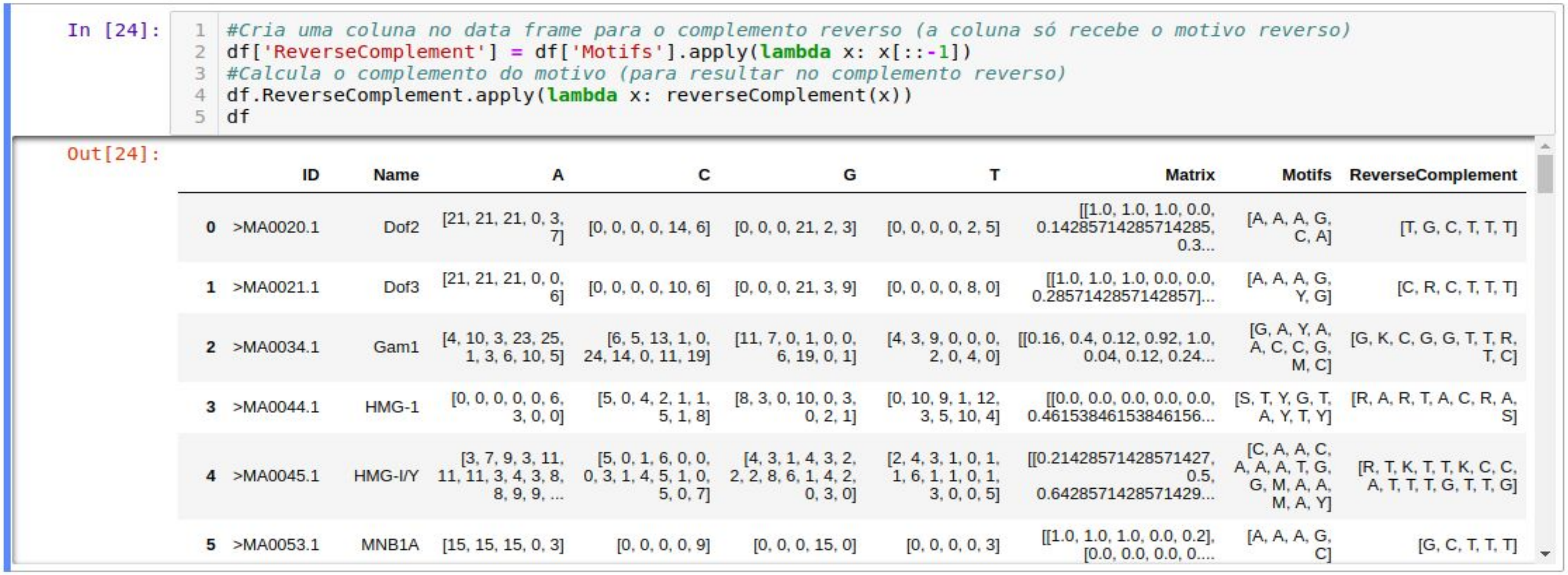
Solução



Gray code	Base	IUPAC	Gray code	Base	IUPAC
0000	-	#	1100	A|C	M
0001	T	T	1101	A|C|T	H
0011	G|T	K	1111	A|C|G|T	N
0010	G	G	1110	A|C|G	V
0110	C|G	S	1010	A|G	R
0111	C|G|T	B	1011	A|G|T	D
0101	C|T	Y	1001	A|T	W
0100	C	C	1000	A	A

Modelo de consenso da IUPAC (International Union of Pure and Applied Chemistry)

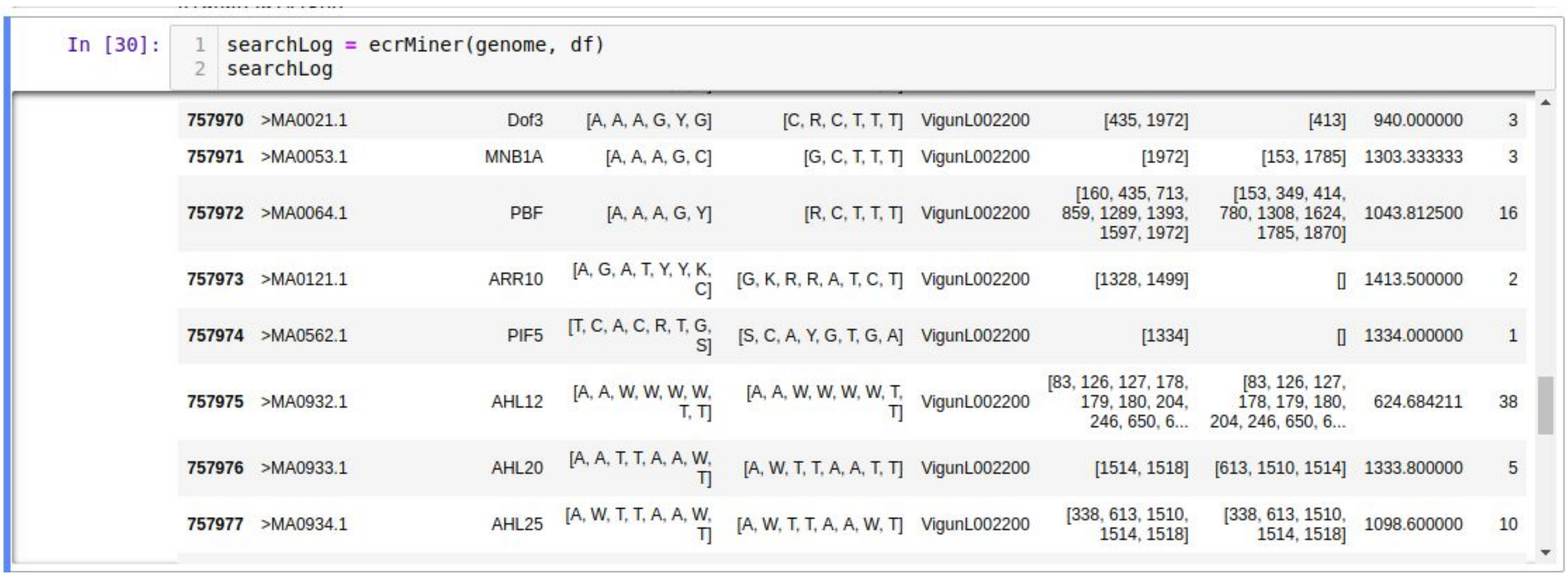
Aplicando ao novo modelo



```python
#Cria uma coluna no data frame para o complemento reverso (a coluna só recebe o motivo reverso)
df['ReverseComplement'] = df['Motifs'].apply(lambda x: x[::-1])
#Calcula o complemento do motivo (para resultar no complemento reverso)
df.ReverseComplement.apply(lambda x: reverseComplement(x))
df
```

	ID	Name	A	C	G	T	Matrix	Motifs	ReverseComplement
0	>MA0020.1	Dof2	[21, 21, 21, 0, 3, 7]	[0, 0, 0, 0, 14, 6]	[0, 0, 0, 21, 2, 3]	[0, 0, 0, 0, 2, 5]	[[1.0, 1.0, 1.0, 0.0, 0.14285714285714285, 0.3...	[A, A, A, G, C, A]	[T, G, C, T, T, T]
1	>MA0021.1	Dof3	[21, 21, 21, 0, 0, 6]	[0, 0, 0, 0, 10, 6]	[0, 0, 0, 21, 3, 9]	[0, 0, 0, 0, 8, 0]	[[1.0, 1.0, 1.0, 0.0, 0.0, 0.2857142857142857]...	[A, A, A, G, Y, G]	[C, R, C, T, T, T]
2	>MA0034.1	Gam1	[4, 10, 3, 23, 25, 1, 3, 6, 10, 5]	[6, 5, 13, 1, 0, 24, 14, 0, 11, 19]	[11, 7, 0, 1, 0, 0, 6, 19, 0, 1]	[4, 3, 9, 0, 0, 0, 2, 0, 4, 0]	[[0.16, 0.4, 0.12, 0.92, 1.0, 0.04, 0.12, 0.24...	[G, A, Y, A, A, C, C, G, M, C]	[G, K, C, G, G, T, T, R, T, C]
3	>MA0044.1	HMG-1	[0, 0, 0, 0, 0, 6, 3, 0, 0]	[5, 0, 4, 2, 1, 1, 5, 1, 8]	[8, 3, 0, 10, 0, 3, 0, 2, 1]	[0, 10, 9, 1, 12, 3, 5, 10, 4]	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.46153846153846156...	[S, T, Y, G, T, A, Y, T, Y]	[R, A, R, T, A, C, R, A, S]
4	>MA0045.1	HMG-I/Y	[3, 7, 9, 3, 11, 11, 11, 3, 4, 3, 8, 8, 9, 9, ...	[5, 0, 1, 6, 0, 0, 0, 3, 1, 4, 5, 1, 0, 5, 0, 7]	[4, 3, 1, 4, 3, 2, 2, 2, 8, 6, 1, 4, 2, 0, 3, 0]	[2, 4, 3, 1, 0, 1, 1, 6, 1, 1, 0, 1, 3, 0, 0, 5]	[[0.21428571428571427, 0.5, 0.6428571428571429...	[C, A, A, C, A, A, A, T, G, G, M, A, A, M, A, Y]	[R, T, K, T, T, K, C, C, A, T, T, T, G, T, T, G]
5	>MA0053.1	MNB1A	[15, 15, 15, 0, 3]	[0, 0, 0, 0, 9]	[0, 0, 0, 15, 0]	[0, 0, 0, 0, 3]	[[1.0, 1.0, 1.0, 0.0, 0.2], [0.0, 0.0, 0.0, 0....	[A, A, A, G, C]	[G, C, T, T, T]

Pré-processamento

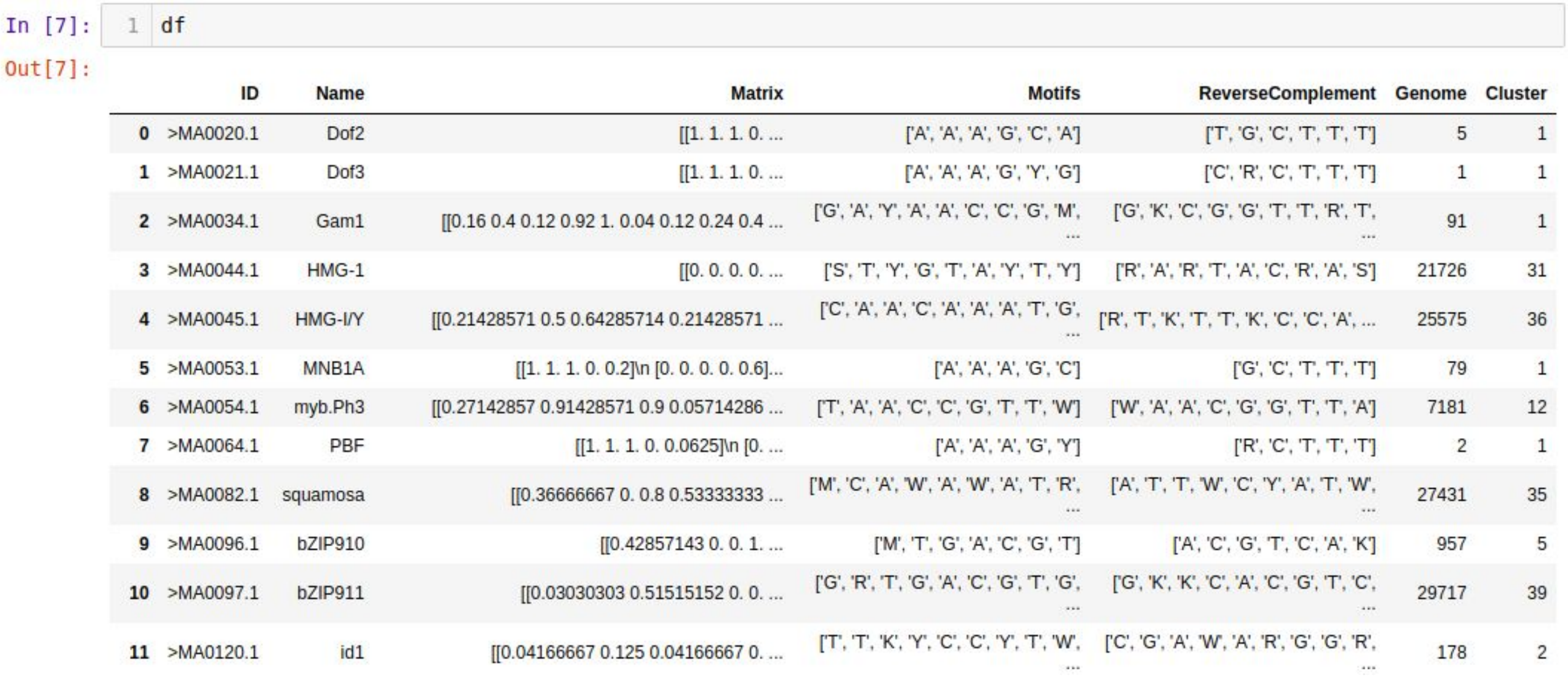


In [30]:

```
searchLog = ecrMiner(genome, df)
searchLog
```

757970	>MA0021.1	Dof3	[A, A, A, G, Y, G]	[C, R, C, T, T, T]	VigunL002200	[435, 1972]	[413]	940.000000	3
757971	>MA0053.1	MNB1A	[A, A, A, G, C]	[G, C, T, T, T]	VigunL002200	[1972]	[153, 1785]	1303.333333	3
757972	>MA0064.1	PBF	[A, A, A, G, Y]	[R, C, T, T, T]	VigunL002200	[160, 435, 713, 859, 1289, 1393, 1597, 1972]	[153, 349, 414, 780, 1308, 1624, 1785, 1870]	1043.812500	16
757973	>MA0121.1	ARR10	[A, G, A, T, Y, Y, K, C]	[G, K, R, R, A, T, C, T]	VigunL002200	[1328, 1499]	[]	1413.500000	2
757974	>MA0562.1	PIF5	[T, C, A, C, R, T, G, S]	[S, C, A, Y, G, T, G, A]	VigunL002200	[1334]	[]	1334.000000	1
757975	>MA0932.1	AHL12	[A, A, W, W, W, W, T, T]	[A, A, W, W, W, W, T, T]	VigunL002200	[83, 126, 127, 178, 179, 180, 204, 246, 650, 6...	[83, 126, 127, 178, 179, 180, 204, 246, 650, 6...	624.684211	38
757976	>MA0933.1	AHL20	[A, A, T, T, A, A, W, T]	[A, W, T, T, A, A, T, T]	VigunL002200	[1514, 1518]	[613, 1510, 1514]	1333.800000	5
757977	>MA0934.1	AHL25	[A, W, T, T, A, A, W, T]	[A, W, T, T, A, A, W, T]	VigunL002200	[338, 613, 1510, 1514, 1518]	[338, 613, 1510, 1514, 1518]	1098.600000	10

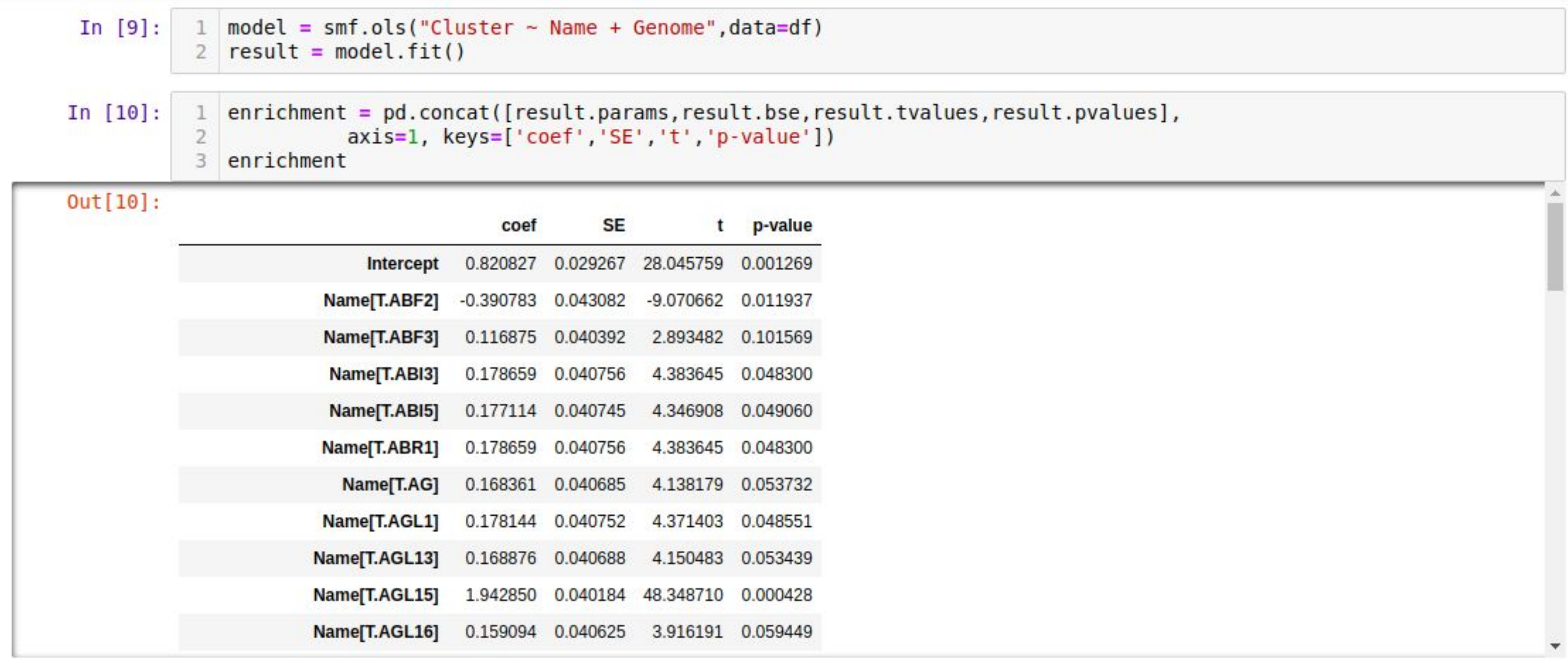
Processamento



```
In [7]: df
Out[7]:
```

	ID	Name	Matrix	Motifs	ReverseComplement	Genome	Cluster
0	>MA0020.1	Dof2	[[1. 1. 1. 0. ...	['A', 'A', 'A', 'G', 'C', 'A']	['T', 'G', 'C', 'T', 'T', 'T']	5	1
1	>MA0021.1	Dof3	[[1. 1. 1. 0. ...	['A', 'A', 'A', 'G', 'Y', 'G']	['C', 'R', 'C', 'T', 'T', 'T']	1	1
2	>MA0034.1	Gam1	[[0.16 0.4 0.12 0.92 1. 0.04 0.12 0.24 0.4 ...	['G', 'A', 'Y', 'A', 'A', 'C', 'C', 'G', 'M', ...	['G', 'K', 'C', 'G', 'G', 'T', 'T', 'R', 'T', ...	91	1
3	>MA0044.1	HMG-1	[[0. 0. 0. 0. ...	['S', 'T', 'Y', 'G', 'T', 'A', 'Y', 'T', 'Y']	['R', 'A', 'R', 'T', 'A', 'C', 'R', 'A', 'S']	21726	31
4	>MA0045.1	HMG-I/Y	[[0.21428571 0.5 0.64285714 0.21428571 ...	['C', 'A', 'A', 'C', 'A', 'A', 'A', 'T', 'G', ...	['R', 'T', 'K', 'T', 'T', 'K', 'C', 'C', 'A', ...	25575	36
5	>MA0053.1	MNB1A	[[1. 1. 1. 0. 0.2]\n [0. 0. 0. 0. 0.6]...	['A', 'A', 'A', 'G', 'C']	['G', 'C', 'T', 'T', 'T']	79	1
6	>MA0054.1	myb.Ph3	[[0.27142857 0.91428571 0.9 0.05714286 ...	['T', 'A', 'A', 'C', 'C', 'G', 'T', 'T', 'W']	['W', 'A', 'A', 'C', 'G', 'G', 'T', 'T', 'A']	7181	12
7	>MA0064.1	PBF	[[1. 1. 1. 0. 0.0625]\n [0. ...	['A', 'A', 'A', 'G', 'Y']	['R', 'C', 'T', 'T', 'T']	2	1
8	>MA0082.1	squamosa	[[0.36666667 0. 0.8 0.53333333 ...	['M', 'C', 'A', 'W', 'A', 'W', 'A', 'T', 'R', ...	['A', 'T', 'T', 'W', 'C', 'Y', 'A', 'T', 'W', ...	27431	35
9	>MA0096.1	bZIP910	[[0.42857143 0. 0. 1. ...	['M', 'T', 'G', 'A', 'C', 'G', 'T']	['A', 'C', 'G', 'T', 'C', 'A', 'K']	957	5
10	>MA0097.1	bZIP911	[[0.03030303 0.51515152 0. 0. ...	['G', 'R', 'T', 'G', 'A', 'C', 'G', 'T', 'G', ...	['G', 'K', 'K', 'C', 'A', 'C', 'G', 'T', 'C', ...	29717	39
11	>MA0120.1	id1	[[0.04166667 0.125 0.04166667 0. ...	['T', 'T', 'K', 'Y', 'C', 'C', 'Y', 'T', 'W', ...	['C', 'G', 'A', 'W', 'A', 'R', 'G', 'G', 'R', ...	178	2

Análise exploratória



In [9]:

```python
model = smf.ols("Cluster ~ Name + Genome",data=df)
result = model.fit()
```

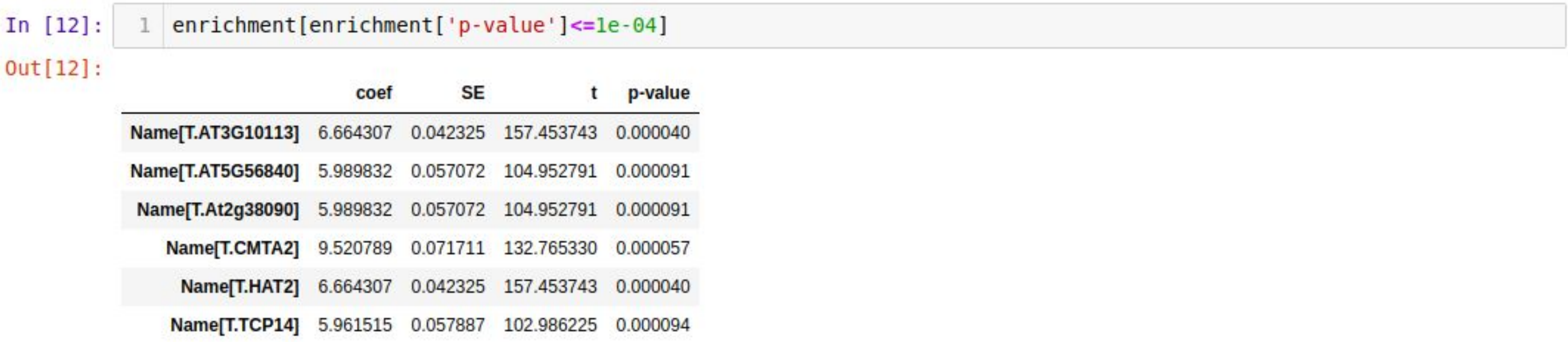
In [10]:

```python
enrichment = pd.concat([result.params,result.bse,result.tvalues,result.pvalues],
          axis=1, keys=['coef','SE','t','p-value'])
enrichment
```

Out[10]:

	coef	SE	t	p-value
Intercept	0.820827	0.029267	28.045759	0.001269
Name[T.ABF2]	-0.390783	0.043082	-9.070662	0.011937
Name[T.ABF3]	0.116875	0.040392	2.893482	0.101569
Name[T.ABI3]	0.178659	0.040756	4.383645	0.048300
Name[T.ABI5]	0.177114	0.040745	4.346908	0.049060
Name[T.ABR1]	0.178659	0.040756	4.383645	0.048300
Name[T.AG]	0.168361	0.040685	4.138179	0.053732
Name[T.AGL1]	0.178144	0.040752	4.371403	0.048551
Name[T.AGL13]	0.168876	0.040688	4.150483	0.053439
Name[T.AGL15]	1.942850	0.040184	48.348710	0.000428
Name[T.AGL16]	0.159094	0.040625	3.916191	0.059449

Análise de enriquecimento



```
In [12]:  enrichment[enrichment['p-value']<=1e-04]
Out[12]:
```

	coef	SE	t	p-value
Name[T.AT3G10113]	6.664307	0.042325	157.453743	0.000040
Name[T.AT5G56840]	5.989832	0.057072	104.952791	0.000091
Name[T.At2g38090]	5.989832	0.057072	104.952791	0.000091
Name[T.CMTA2]	9.520789	0.071711	132.765330	0.000057
Name[T.HAT2]	6.664307	0.042325	157.453743	0.000040
Name[T.TCP14]	5.961515	0.057887	102.986225	0.000094

“

As famílias de elementos cis regulatórios CAMTA, Homeobox, Myb-related e TCP na espécie da planta *Vigna unguiculata* (Feijão macassar) são fortes candidatos a regulação da via metabólica dos fenilpropanóides, responsável pela produção de óleos essenciais, produto largamente utilizado na indústria no controle de fungos, e portanto, de grande interesse biotecnológico.

Muito obrigado!

Github.com/filipecmedeiros
LinkedIn/in/filipecmedeiros