

Origen de replicación

Bioinformática 2025-2

Universidad de Sonora

8 de octubre de 2025

¿Cómo encontrar un gato negro en un cuarto oscuro?

53†††305))6.;4826)4†.)4†);806.;48†8^60))85;161;:†.8
†83(88)5.†;46(;88.96.?.8).†(;485);5.†2: .†(;4956.2(5
.-4)8^8.;4069285);)6†8)4††;1(†9;48081;8:8†1;48†85;4
)485†528806.81(†9;48;(88;4(†?34;48)4†;1†(;:188;†?;

¿Cómo encontrar un gato negro en un cuarto oscuro?

53†††305))6·;**48**26)4†.)4†);806·;**48**†8^60))85;161;:†·8
†83(88)5·†;46(;88·96·?;8)·†(;**48**5);5·†2:·†(;4956·2(5
·-4)8^8·;4069285);)6†8)4††;1(†9;**48**081;8:8†1;**48**†85;4
)485†528806·81(†9;**48**; (88;4(†?34;**48**)4†;1†(;:188;†?;

¿Cómo encontrar un gato negro en un cuarto oscuro?

53†††305))6·THE26)H†.)H†)TE06·THE†E^60))E5T161T:†·E
†E3(EE)5·†TH6(TEE·96·?TE)·†(THE5)T5·†2:·†(TH956·2(5
·-H)E^E·TH0692E5)T)6†E)H††T1(†9THE0E1TE:E†1THE†E5TH
)HE5†52EE06·E1(†9THET(EETH(†?3HTHE)H†T1†(T:1EET†?T

Contar k-meros

PatternCount(*Text*, *Pattern*)

count $\leftarrow 0$

for *i* $\leftarrow 0$ to $|Text| - |Pattern|$

if *Text*(*i*, $|Pattern|$) = *Pattern*

count $\leftarrow count + 1$

return *count*

K-meros más frecuentes

FrequentWords(*Text*, *k*)

FrequentPatterns \leftarrow an empty set

Count \leftarrow an array of length $|Text| - k + 1$

for *i* $\leftarrow 0$ to $|Text| - k$

Pattern $\leftarrow Text(i, k)$

Count(*i*) \leftarrow **PatternCount**(*Text*, *Pattern*)

maxCount \leftarrow maximum value in array *Count*

for *i* $\leftarrow 0$ to $|Text| - k$

if *Count*(*i*) = *maxCount*

 add *Text*(*i*, *k*) to *FrequentPatterns*

remove duplicates from *FrequentPatterns*

return *FrequentPatterns*

Algo más eficiente...

ACG	CGT	GTT	TTT	TTC	TCA	CAC	TTA	TAC	CGG
3	2	2	3	1	1	1	1	1	1

FrequencyTable

```
FrequencyTable(Text, k)  
  freqMap  $\leftarrow$  empty map  
  n  $\leftarrow$  |Text|  
  for i  $\leftarrow$  0 to n - k  
    Pattern  $\leftarrow$  Text(i, k)  
    if freqMap[Pattern] doesn't exist  
      freqMap[Pattern]  $\leftarrow$  1  
    else  
      freqMap[pattern]  $\leftarrow$  freqMap[pattern]+1  
  return freqMap
```


BetterFrequentWords

BetterFrequentWords(*Text*, *k*)

FrequentPatterns \leftarrow an array of strings of length 0

freqMap \leftarrow **FrequencyTable**(*Text*, *k*)

max \leftarrow **MaxMap**(*freqMap*)

for all strings *Pattern* in *freqMap*

if *freqMap*[*pattern*] = *max*

 append *Pattern* to *frequentPatterns*

return *frequentPatterns*