

DNA file sequence & DNA ^{and} marker CSV

DNA → sequ. of molecules - nucleotides

Each nucleotide:

either

Background notes

adenine (A)

cytosine (C)

guanine (G)

thymine (T)

Short Tandem Repeats (STR)

consecutive nucleotide structure

of STR repeats varies

Multiple STRs improve accuracy

FBI uses 20 STRs

Species `argv[1] → csv file w/ STRs`

`argv [2] → ta file w/ identity`

`re.search (STR, list[0])`

DNA

Pseudo Code

1. open CSV file / load into memory
2. open txt file / load into memory
3. Search txt file for each pattern
4. Record # of times pattern is found
5. Compare those numbers w/ numbers from CSV file, now you can use if to convert it to a list or dictionary to work with entries
6. If complete match print name at the index of shortener
7. If No match print no match

Step 4

loop

for key in shortener[0]

if none - skip

find and matches of key in one store in list (count)

Now iterate over count

dict w/
+ ST Rice
keys

Step 5 Compare values in Cont
w/ values in short tan

loop

for i in range(len(cont))

(Key){1, len(sir(dna))}, dna

We need to find a method
for isolating the longest sequence
of repeated STRs and
the number of repetitions

ABB A A (BBB) AAA BBBB AA

Search repeat of B

result 4

nd = X for x in range(len(dna)) if
dna.startswith(key, x)

print(nd)

Handle over ind?

Revise Step 4

↓
4. Record number of occurrences in
longest chain of repetitions

i.e. aabbccaaabbb

group → 3

using list comprehension and `startswith`
we can pinpoint the indices
of occurrences in dna string

using string

of ABs ⇒ [1, 2, 3, 4, 8, 11, 12, 13]

on previous

page

given B

Here's over there, list
using conditional to find
longest chain of repetitions
list assigned to ind

for i in range(len(ind))

for i in range(1, len(ind) - 1):
 if ind[i] == ind[i + 1]:
 if len(ind[i:i + 2]) > max_length:
 max_length = len(ind[i:i + 2])
 max_index = i

Doesn't work w/ longer
substrings, e.g. not ((11)-111)

Alternative use while loop w/ nested
for loop (further)

- vice versa actually

where mid
is 11st
of indices

for j in range(mid): erg

for i in range(len(dna) - 1: len(key))

len(dna[int(j): len(key)])

113, 118, 123

o 1 loop

52

BAC

- 57

5

Key = "BAC"

Count = 0 countf = 0

string = [A BAC AA BB BAC BAC BACAA]

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

0 12 ... 13 ... 148 ...]

dna

1st comp + starts with

List = [1, 8, 11, 14]

13 14 15 16 17 18

for i in range(len(string)):

if string[i] == key: countf += 1

for j in range(tmp, len(string), len(key))

if string[j:j+len(key)] == key: count += 1

count += 1

else

break

if count > countf:

countf = count

Step 5 Compare values returned by count to values in database

strcount [3, 7, 9, 10]
tmpList = []

database [0] { a, b, c, d }
[1] { }
[2] { }

for ii in range(len(database))

 for key in database[ii]

 if name

 count

 else key

 tmpList.append(database[ii][key])

 if tmpList == strCount

 print match

sys.exit(0)