Esame 18/07/2014

Andrea Passerini
passerini@disi.unitn.it

Informatica

Programma python

Scrivere una funzione `computeBindingStats(filename)` che:

- prenda in ingresso un nome di file `filename` con un elenco di siti di binding di RBP su UTR
- stampi per ciascuna RBP, numero medio e lunghezza media dei siti di binding da essa legati, e numero medio e lunghezza media dei siti di binding di ciascun gene (considerando tutti i suoi UTR)

Esempio di input

```
>>> cat "bindings.txt"
idGene geneSymbol idUTR name idRegulatoryElement idRegulatoryElementType elementName evidence
29305 ZRANB2 67123 uc001dfs.2_3UTR 1 RBP IGF2BP3 PAR-CLIP 1460 1738 20371350
3977 CDC42SE1 68361 uc001ewp.2_3UTR 10 RBP IGF2BP3 PAR-CLIP 922 954 20371350
27461 UBE2J2 64625 uc001adq.2_3UTR 100 RBP IGF2BP3 PAR-CLIP 1088 1160 20371350
25732 TMEM201 64976 uc001apy.2_3UTR 1000 RBP IGF2BP1 PAR-CLIP 2550 2598 20371350
18974 PTBP2 67480 uc001dru.2_3UTR 10000 RBP IGF2BP1 PAR-CLIP 2149 2216 20371350
16097 OGT 126057 uc004ead.2_3UTR 100000 RBP IGF2BP3 PAR-CLIP 1968 2033 20371350
16097 OGT 126057 uc004ead.2_3UTR 100003 RBP AGO1 PAR-CLIP 1988 2016 20371350
189 ACRC 126058 uc004eae.2_3UTR 100004 RBP AGO1 PAR-CLIP 78 98 20371350
17837 PIN4 126068 uc004eam.2_3UTR 100005 RBP IGF2BP1 PAR-CLIP 236 284 20371350
17837 PIN4 126068 uc004eam.2_3UTR 100006 RBP IGF2BP3 PAR-CLIP 239 270 20371350
17837 PIN4 126068 uc004eam.2_3UTR 100007 RBP IGF2BP3 PAR-CLIP 272 362 20371350
17837 PIN4 126068 uc004eam.2_3UTR 100008 RBP IGF2BP2 PAR-CLIP 295 368 20371350
17837 PIN4 126068 uc004eam.2_3UTR 100009 RBP IGF2BP1 PAR-CLIP 332 366 20371350
18974 PTBP2 67490 uc001drt.2_3UTR 10001 RBP IGF2BP3 PAR-CLIP 563 614 20371350
17837 PIN4 126068 uc004eam.2_3UTR 100010 RBP IGF2BP1 PAR-CLIP 441 473 20371350
17837 PIN4 126068 uc004eam.2_3UTR 100011 RBP IGF2BP2 PAR-CLIP 441 481 20371350
6949 ERCC6L 63047 uc004eap.1_5UTR 100016 RBP AGO1 PAR-CLIP 504 527 20371350
21631 RPS4X 126073 uc011mqb.1_3UTR 100017 RBP IGF2BP1 PAR-CLIP 0 36 20371350
21631 RPS4X 126073 uc011mqb.1_3UTR 100018 RBP IGF2BP1 PAR-CLIP 57 171 20371350
...
```

Esempio esecuzione

```
>>> import utility
>>> utility.computeBindingStats('bindings.txt')
reg_name          num_binds       avg_length
QKI       1135    31.77
PUM2      5740    35.30
TNRC6C    288     32.14
TNRC6B    1244    26.58
TNRC6A    401     29.02
IGF2BP1 38876     62.63
AGO4      429     26.09
IGF2BP3 64800     70.31
IGF2BP2 45271     63.03
AGO1      4428    32.47
AGO3      51      31.45
AGO2      318     24.94
gene_name         num_binds       avg_length
UBE2Q1    21      79.86
RNF14     25      55.44
UBE2Q2    14      46.71
RNF10     6       69.50
RNF11     48      59.42
RNF13     6       49.83
PMM2      19      78.21
...
```

Programma python: suggerimento

Si possono implementare 4 funzioni separate:

1. una che legga il file `filename` e restituisca una lista di tuple (nome del gene, nome dell'RBP, inizio sito, fine sito)
2. una che prenda in ingresso la lista di tuple, e restituisca due dizionari: uno da nome gene a lista di siti (inizio sito, fine sito) e uno da nome RBP a lista di siti (inizio sito, fine sito)
3. una che prenda in ingresso un dizionario da nome a lista di siti, e stampi per ogni nome numero e lunghezza media dei suoi siti di binding
4. una che realizzi il programma richiesto usando le funzioni di cui sopra

Shell: esercizio #1

Dato il file `proteins.fasta`, calcolare quante di queste:

1. Contengono il motivo: una fenilalanina (F), seguita da un acido aspartico (D), seguito da una fenilalanina; oppure un aminoacido qualunque, seguito da un acido aspartico, seguito da un triptofano (W).
2. Contengono il motivo: una arginina (R) o lisina (K), seguita da due amino acidi qualunque, seguiti da una arginina o lisina.
3. Contengono il primo motivo ma non il secondo.

Soluzione

(1) 18. (2) 126. (3) 9.

Shell: esercizio #2

Controllare se il file `proteins.fasta` contiene almeno due catene diverse di una stessa proteina.

Spiegazione

Gli identificativi `"2qfd:g"` e `"2qfd:h"` si riferiscono alla proteina `"2qfd"` ed a due sue catene `"g"` ed `"h"`.

Soluzione

No.

Modalita' di esecuzione e consegna

1. Avviare la macchina in modalita' `ESAME`
2. Autenticarsi con nome utente `sci-esame` e password fornita dal docente
3. Il testo del compito ed i file necessari si trovano in una cartella `Testo` sul Desktop
4. Realizzare il programma python come file `utility.py` e scrivere gli esercizi da linea di comando in un file di testo `linea_di_comando.txt`
5. Creare sul Desktop una cartella con *nome_cognome* e metterci i due file realizzati.
6. Eseguire il logout ma NON spegnere la macchina