

Esame 12/02/2015

Andrea Passerini
passerini@disi.unitn.it

Informatica

Programma python

Scrivere un programma python che:

- prenda in ingresso un nome di file `filename` che contiene annotazioni di una sequenza in formato UNIPROT
- stampi per ciascun elemento di struttura secondaria (HELIX,STRAND,TURN) la lunghezza media con cui appare nella sequenza, ed il numero medio di volte consecutive con cui appare nella sequenza.

Esempio ingresso

```
ID      ZN224_HUMAN                      Reviewed;              707 AA.
AC      Q9NZL3; A6NFW9; P17033; Q86V10; Q8IZC8; Q9UID9; Q9Y2P6;
DT      08-DEC-2000, integrated into UniProtKB/Swiss-Prot.
DT      30-NOV-2010, sequence version 3.

....
....
....
FT                               {ECO:0000305}.
FT      CONFLICT      562      562      H -> R (in Ref. 1; AAF04106).
FT                               {ECO:0000305}.
FT      STRAND        174      177      {ECO:0000244|PDB:2EN8}.
FT      TURN          179      181      {ECO:0000244|PDB:2EN8}.
FT      STRAND        184      187      {ECO:0000244|PDB:2EN8}.
FT      HELIX         188      195      {ECO:0000244|PDB:2EN8}.
FT      TURN          196      198      {ECO:0000244|PDB:2EN8}.
FT      STRAND        207      209      {ECO:0000244|PDB:2EM6}.
FT      HELIX         216      223      {ECO:0000244|PDB:2EM6}.
FT      TURN          224      226      {ECO:0000244|PDB:2EM6}.
FT      STRAND        227      229      {ECO:0000244|PDB:2EM6}.
FT      STRAND        235      237      {ECO:0000244|PDB:2ELY}.
FT      STRAND        241      243      {ECO:0000244|PDB:2ELY}.

....
```

Esempio esecuzione

```
python program.py
```

```
Data file: Q9NZL3.txt
```

```
lunghezze medie
```

```
{'TURN': 3.125, 'HELIX': 7.285714285714286,  
  'STRAND': 3.323529411764706}
```

```
occorrenze consecutive medie
```

```
{'TURN': 1.0, 'HELIX': 1.4, 'STRAND': 1.736842105263158}
```

Programma python: suggerimento (1)

Si possono implementare 5 funzioni separate:

- 1 una che legga il file e restituisca una lista di coppie: elemento di struttura secondaria (ss) - lunghezza (NB: entrambi gli estremi vanno considerati)
- 2 una che prenda in ingresso la lista letta, e costruisca due dizionari, entrambi con ss come chiave, e come valore: il primo la somma delle sue lunghezze, il secondo il numero di volte in cui compare. I due dizionari saranno utilizzati per ottenere un dizionario di lunghezze normalizzato (usando una funzione ausiliaria), che la funzione deve restituire
- 3 una che prenda in ingresso due dizionari e restituisca un nuovo dizionario ottenuto normalizzando i valori del primo con quelli del secondo

Programma python: suggerimento (2)

- 4 una che prenda in ingresso la lista letta, e costruisca due dizionari, entrambi con ss come chiave, e come valore: il primo il numero di volte in cui compare, il secondo il numero di volte *non consecutive* in cui compare. La funzione restituirà il dizionario di occorrenze normalizzato
- 5 una che realizzi il programma richiesto usando le funzioni di cui sopra

Shell: esercizio #1

Dato i file nella directory *fasta*, calcolare quante sequenze contengono il primo, il secondo o almeno uno di questi motivi:

- 1 Un acido aspartico (D); opzionalmente una serina (S); una glicina (G); due o tre aminoacidi qualunque; una serina oppure una treonina (T).
- 2 Una asparagina (N) opzionale, un aminoacido che non sia una prolina (P), una cisteina (C).

Soluzione

Il primo: 14; Il secondo 112; Almeno uno dei due: 116.

Shell: esercizio #2

Ogni riga del file `interactions.csv` rappresenta una interazione tra due domini proteici, riportati nella prima e terza colonna. Trovare i tre domini che compaiono più volte, indipendentemente dalla colonna in cui si trovano.

Input

| family | identifier | family | identifier | interaction |
|---------|-----------------|---------|--------------|-------------|
| PF02455 | Hex_IIIa | PF01310 | Adeno_PVIII | interchain |
| PF08977 | BOFC_N | PF08955 | BofC_C | intrachain |
| PF04898 | Glu_syn_central | PF01645 | Glu_synthase | intrachain |
| ... | | | | |

Soluzione

```
60 PF00129
64 PF00071
71 PF07686
```


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