

## Programma python

Scrivere una funzione `averageMatchConservation(data)` che:

- prenda in ingresso un nome di file `data` che contiene un allineamento multiplo di domini proteici
- stampi i valori di conservazione di ciascuna posizione di match

## File degli allineamenti (alignments)

```
# STOCKHOLM 1.0
#=GF ID   RRM_1
#=GF AC   PF00076.17
#=GF DE   RNA recognition motif. (a.k.a. RRM, RBD, or RNP domain)
#=GF PI   rrm;
#=GF AU   Eddy SR, Birney E
....
AOCEN5_PARTE/256-312  [...] .kemr-----.[...]
AOCA51_PARTE/258-315  [...] ..ekemr-----.....-..[...]
AOC441_PARTE/141-196  [...] ....T...Y.....G.....P.[...]
A0BF98_PARTE/253-307  [...] .....T...Y.....G.....P...[...]
```

## Esempio esecuzione

```
>>> import utility
>>> utility.averageMatchConservation('alignments')
(0, 0.8163457612212206)
(1, 0.8800138203976505)
(2, 0.9138737946414549)
(3, 0.9218519332851713)
(4, 0.9263749725162547)
(5, 0.9320915915444294)
(6, 0.9345101611332726)
(7, 0.9389703803750353)
(8, 0.9449068693658322)
(9, 0.9511574583032321)
(10, 0.955429217577033)
....
```

## Programma python: suggerimento

Si possono implementare 5 funzioni separate:

1. una che legga il file e restituisca un dizionario con nome di dominio proteico come chiave, e suo allineamento come valore
2. una che dato un allineamento, restituisca una lista con 1 per le posizioni di match, e 0 per quelle di cancellazione (ignorando gli inserimenti)
3. una che scorra il dizionario di allineamenti, per ciascuno calcoli la lista di match e cancellazioni (funzione precedente), e calcoli la media di queste liste
4. una che stampi la media di match-cancellazioni
5. una che realizzi il programma richiesto usando le funzioni di cui sopra

**Esercizio Shell 1**

Calcolare il numero di domini proteici nel file `alignments` che cominciano con un inserimento

*Soluzione*

5620

**Esercizio Shell 2**

Contare tra i file `fasta` quante catene:

- cominciano con una alanina (A), seguita da un residuo qualunque e poi da una cisteina (C) o una istidina (H)
- e non finiscono con un acido (E o D)

Esempio: `AGCIKNGGRCNASAGPPYCCSSYCFQIAGQSYGVCKN`

*Soluzione*

6