Esame 22/01/2015

Andrea Passerini
passerini@disi.unitn.it

Informatica

Programma python

Scrivere un programma python che:

- prenda in ingresso un nome di file `filename` che contiene gli allineamenti multipli di una sequenza
- stampi per ciascuna posizione nella sequenza, il profilo di allineamento, ossia quante volte ciascuno dei possibili residui/nucleotidi/gaps e' stato trovato nell'allineamento in quella posizione

Esempio ingresso

```
0.012   C   ---------CCAHSLVVTEASAD-----------------------------------------------
0.013   G   ---------GGGYFAVFATAFSS-----------------------------------------------
0.024   V   ---------VILINLLLYVIIIE-------VI-------------------------------------
0.014   P   ---------PPPCKRGKG-SEDS-------RV-------------------------------------
0.013   A   ---------AKACLSSVY-VIN--------FT------L-----------------------------
0.014   I   ---------IIPPEY-QLLVFI--------HL------P-----------------------------
0.019   Q   ---------QEKENEEGTQGGE--------LG------L-----------------------------
0.020   P   ---------PPTPPSSQKP-DI--------SV------P------A-------------------------
0.077   V   --N------VKAP-PPREESTPVGAVKGTCR-------P--L---SR-E----L-DC---AE-W-S---MQ-
0.087   L   --A------LLLP-VISESPKEDHSPPPSSP-------L--E---FD-IG---WDRG---EA-T-T---QI-
0.106   S   --K------SPPL-GSEERIPVTSSSGETGF-------N--KA--ADDQT-F-RLLW--KHP-G-P---TQ-
0.103   G   --T------GGQY-V-PIQKPDNIRFFQTSF-------DG-LT--DSRGA-Y-TDSD--AVD-R-Q---VA-
0.202   L   --F------LAA--RQLQSDSSGPATEMSHG-QQQSSSRD-SGQSKTRNANV-HQMPV-LRGLH-Q--NEGS
0.274   S   VVI-E-PPTSNE--EFYSSNVLLPKGGGPSA-GGGFLSEGPGEENPNSDSGR-ASASGPKSFAP-P-DENQT
0.399   R   GGR-R-RRRRRRR-QKRRRRKER-RFRRTRR-RRRATSRKKRERGDMRRERS-RRLRRDELETR-SAALKDW
0.610   I   IVI-I-IIIIIIM-IIVIVIIIVIIVIVRII-IIIILIIIIIIIIVIIIIII-IIIIVHYVITI-IYLVIVL
0.463   V   YAV-V-VIVVVVAVLLIVIVVVVIVIIVIIV-LMAHQRVVVVIVYAVRVVVV-IIIVAINMVVYPISHLRII
0.506   N   GGG-G-GGGNGGYEGGNGNSGNNEGNNGNCG-GGGGGNGNGGNGNNGEGGGG-GGGNDANGGSNLGSRGHGR
0.794   G   GGG-G-GGGGGGGAGGGGGGGGGGGGGGGGG-GGGGDGGGGGGGGGGGGGGG-GGGGGGSGGTGGGSGGGGS
0.288   E   HHQ-S-HHKESITSHHKSRSSQQTDTHSGQS-EEETRVSTSEKVVGMQ-RIH-TYTVS-YGQVIK-SITPVP
0.323   E   DED-Y-PVEEDEASNNPSDDPENEEEENQNV-DDDKKRNTEVKEEDKL-RDA-EDDED-LTKIKE-PPELSV
0.493   A   IIA-I-SSTAASANAAASAAAVAAAAAARAAAVAAIITAAATAVAAVARAAS-AVVTA-PVASTCSAVVTAQ
0.320   V   SPN-P-DSTVVTRLDPDSPKRVNNAEEAVKSEADVDERSGEPKPKADKPRVLMNDEER-EPQFTPNRQPSNH
0.351   P   IIIII-VIIPPIPVIVLVPLLPRLPPKVNKEVQALIIIPPRKKIFEIDSPAPTIIDIR-NVFRVPAIQVIDR
0.372   G   ESQEE-WKSGHNNHKAFGGGHHGHHHGNENGGGTGKTYGGNFNHDNEDGHGGRSEGNGTFGGPEHTEQGVIT
0.388   S   QDDQY-HEESSQEQLSSQSQQSQEESESTSQSEAQDDLDEEQSLFSQEGADAHNDKEQTNARRSSSDNKKIE
0.347   W   AAYAV-QEVWYWYNHANFFFFIFFFAAAVAFQYTLAAAAFMYFTWFVFYWAAWFAAFFPVKHYLRGYYYYAF
0.739   P   PPPPP-PKPPPSPPPPPPKPPPPPPPPPPPPPTPPPPVPPPQPTPPTSPPPPPPPPPPPTRPGPPPGPPRP
0.521   W   FFYWW-HYYWWSWWWWWYYWWYYYYYFYFYHHWFYWWYYFYWYWFYYGFFWYWWYYMYYYYYYGWWSYFFSF
....
```

Esempio esecuzione

```
python program.py
Data file: alignment
el   -   A   C   D   E   F   G   H   I   K   L   M   N   P   Q   R   S   T   V   W   X   Y
C   58   3   2   1   1   0   0   1   0   0   1   0   0   0   0   0   2   1   2   0   0   0
G   58   3   0   0   0   3   3   0   0   0   0   0   0   0   0   0   2   1   1   0   0   1
V   56   0   0   0   1   0   0   0   6   0   4   0   1   0   0   0   0   0   3   0   0   1
P   57   0   1   1   1   0   2   0   0   2   0   0   0   3   0   2   2   0   1   0   0   0
A   57   2   1   0   0   1   0   0   1   1   2   0   1   0   0   0   2   1   2   0   0   1
I   57   0   0   0   1   1   0   1   3   0   3   0   0   3   1   0   0   0   1   0   0   1
Q   56   0   0   0   5   0   4   0   0   1   2   0   1   0   2   0   0   1   0   0   0   0
P   56   1   0   1   0   0   0   0   1   1   0   0   0   6   1   0   3   1   1   0   0   0
V   36   3   2   1   4   0   2   0   0   2   2   1   1   5   1   3   3   2   3   1   0   0
....
```

Attenzione!

Non tutte le righe contengono tutte le possibili sostituzioni (si vedano i molti zeri nella tabella). E' necessario estrarre un alfabeto (quello riportato nell'intestazione dell'output) con tutte le sostituzioni possibili, da usare come riferimento.

Programma python: suggerimento

Si possono implementare 5 funzioni separate:

1. una che legga il file `filename` e restituisca una lista di coppie residuo-allineamento
2. una che prenda in ingresso la lista letta da file, e restituisca una lista ordinata con l'alfabeto usato nell'allineamento ('-' incluso)
3. una che prenda in ingresso una stringa con un allineamento, e restituisca un profilo (dizionario da carattere a numero di occorrenze nell'allineamento)
4. una che prenda in ingresso lista letta da file e alfabeto, e stampi un'intestazione e per ogni posizione nella lista, l'elemento corrispondente ed il profilo (calcolato chiamando funzione 3) ordinato come da alfabeto (se una lettera non e' presente nel profilo, mettere zero)
5. una che realizzi il programma richiesto usando le funzioni di cui sopra

Shell: esercizio #1

Dati i file fasta nella directory `fasta`, calcolare quante sequenze contengono almeno uno dei seguenti motivi:

1. Un triptofano (W); *tre* aminoacidi qualunque; una fenilalanina (F) o una tirosina (Y). Il motivo può trovarsi in una posizione qualunque.
2. Un aminoacido qualunque; una isoleucina (I) oppure una leucina (L) oppure una metionina (M); una arginina (R). Il motivo deve trovarsi alla *fine* della sequenza.

Ci sono sequenze che li contengono entrambi? (Può essere dedotto dai casi precedenti.)

Soluzione

(1) 14. (2) No.

Shell: esercizio #2

Dato il file `elm_classes.txt`, contenente informazioni su motivi proteici (il loro ID, il nome scientifico, la probabilità del motivo, e il numero di istanze note), trovare gli *ID* dei cinque motivi con più istanze note assieme la loro *probabilità*.

Contenuti

```
# Accession ELMIdentifier Probability #Instances
ELME000321 CLV_C14_Caspase3-7 0.00309374033071 39
ELME000172 CLV_MEL_PAP_1 0.000105375572148 12
...
```

Soluzione

```
ELME000190 0.000119206092212
ELME000056 5.86028045872e-05
ELME000104 1.33321386458e-07
ELME000136 0.015433212802
ELME000070 0.00501781699892
```

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