CAT-PAC4-enun

December 14, 2020

1 Programació per a la ciència de dades - PAC4

En aquest Notebook trobareu un exercici que suposa la quarta activitat d'avaluació continuada (PAC) de l'assignatura. Aquesta PAC consta d'un únic exercici a resoldre, que engloba molts dels conceptes coberts durant l'assignatura.

L'objectiu d'aquest exercici és desenvolupar un **paquet de Python**, fora de l'entorn de Notebooks, que ens permeti resoldre el problema donat. Aquest ha d'incloure el corresponent codi organitzat lògicament (separat en mòduls, organitzats per funcionalitat), la documentació del codi (docstrings) i tests. A més, s'han d'incloure els corresponents arxius de documentació d'alt nivell (README), així com els arxius de llicència i dependències (`requirements.txt`).

Se'ns demana que implementem un paquet de Python que sigui capaç de realitzar una anàlisi de dades amb informació genètica sobre el Bacil de Koch. En particular, ens centrarem en les *pautes obertes de lectura* dels gens del Bacil de Koch. Les *pautes obertes de lectura* són més conegudes pel seu acrònim *ORF* que ve del seu nom en anglès *open reading frame*. L'ORF és una seqüència de nucleòtids que potencialment pot codificar una proteïna. En els organismes eucariotes, com som els éssers humans, cada gen té un únic ORF. Però aquest no és el cas dels bacteris, que són organismes procariotes.

2 Les dades

Les dades a analitzar ens són proporcionades en dues col · leccions de dades separades: `tb_functions.pl` i `orfs/tb_data_0X.pl`. Aquestes dades provenen del repositori UCI Machine Learning i tenen format de *datalog*.

`tb_functions.pl` conté informació general sobre els gens i les seves classes funcionals. Mentre que `tb_data_0X.pl` conté informació detallada sobre tots els gens indicats.

Fent una ullada als arxius proporcionats, podreu veure que els diferents arxius contenen força informació. Per a resoldre l'exercici proposat, només serà necessari fer servir una quantitat molt reduïda d'aquesta.

2.1 tb_functions.pl

L'arxiu `tb_functions.pl` conté informació sobre 123 classes funcionals d'ORFs i té la següent estructura:

```
class([1,0,0,0],"Small-molecule metabolism ").
class([1,1,0,0],"Degradation ").
class([1,1,1,0],"Carbon compounds ").
function(tb186,[1,1,1,0],'bglS',"beta-glucosidase").
function(tb2202,[1,1,1,0],'cbhK',"carbohydrate kinase").
function(tb727,[1,1,1,0],'fucA',"L-fuculose phosphate aldolase").
class([1,1,2,0],"Amino acids and amines ").
function(tb1905,[1,1,2,0],'aao',"D-amino acid oxidase").
function(tb2531,[1,1,2,0],'adi',"ornithine/arginine decarboxylase").
function(tb2780,[1,1,2,0],'ald',"L-alanine dehydrogenase").
function(tb1538,[1,1,2,0],'ansA',"L-asparaginase").
...
```

On hi ha dos tipus d'entrada:

`class`: té 2 elements separats per comes sempre presentats en el següent ordre: * *identificador de la classe*: llista de 4 números que descriu la classe en 4 dimensions diferents (separats per comes i entre claudàtors), i * *descripció de la classe*: string que conté la descripció de la classe, cap classe comparteix descripció amb altra classe (string entre doble cometes).

`function`: té 4 elements separats per comes sempre presentats en el següent ordre: * *ORF*: pauta oberta de lectura (en anglès *open reading frame*) (string sense cometes), * *identificador de la classe*: llista de 4 números que descriu la classe en 4 dimensions diferents (separats per comes i entre claudàtors), * *nom del gen*: nom del gen o valor *null* si el gen no té nom (string entre cometes simples), i * *descripció ORF*: descripció de la pauta de lectura (string entre dobles cometes).

2.2 tb_data_0X.pl

Els arxius `tb_data_0X.pl` tenen la següent estructura:

```
begin(model(tb4)).
tb_protein(tb4).
function(5,0,0,0,'null','null').
coding_region(4434,4994).
tb_mol_wt(19934).
...
sequence_length(187).
amino_acid_pair_ratio(a,a,24.8).
amino_acid_pair_ratio(a,c,0.0).
amino_acid_pair_ratio(a,d,0.0).
amino_acid_pair_ratio(a,e,18.6).
amino_acid_pair_ratio(a,f,0.0).
amino_acid_pair_ratio(a,g,12.4).
...
tb_to_tb_evalue(tb3671,1.100000e-01).
tb_to_tb_evalue(tb405,4.300000e-01).
tb_to_tb_evalue(tb3225,5.600000e-01).
...
species(p35925,'streptomyces_coelicolor').
```

```
classification(p35925,bacteria).
classification(p35925,firmicutes).
classification(p35925,actinobacteria).
classification(p35925,actinobacteridae).
classification(p35925,actinomycetales).
classification(p35925,streptomycineae).
classification(p35925,streptomycetaceae).
classification(p35925,streptomyces).
mol_wt(p35925,19772).
keyword(p35925,'hypothetical_protein').
db_ref(p35925,embl,l27063,g436026,null).
signalip(c,35,no).
signalip(y,35,no).
signalip(s,54,no).
signalip(ss,1,34,no).
signalip(cleavage,null,null).
hydro_cons(-0.498,-0.474,0.624,3.248,0.278).
end(model(tb4)).
begin(model(tb5)).
...
end(model(tb3915)).
```

On les dades per un únic ORF estan capturades entre els delimitadors:

```
begin(model(ORF))
end(model(ORF))
```

I l'atribut `tb_to_tb_evalue(ORF, E-value)` mostra la relació amb altres ORFs.

3 Exercici

Primer, caldrà que llegiu els arxius facilitats de la forma més òptima tenint en compte les tasques demanades i justifiqueu la vostra decisió.

Després, caldrà que genereu funcions que us permetin fer els següents càlculs:

1. Donada la col · lecció de classes funcionals:

   1.1 Calcular quants ORFs pertanyen a cada classe.

   1.2 Donat que el Bacil de Koch afecta sobretot als pulmons, volem que mostreu per pantalla quants ORFs pertanyen a la classe que té *Respiration* com a descripció. Mostreu el resultat per pantalla degudament formatat (utilitzant el mètode `format()` o altre similar), incloent-hi un missatge explicatiu dels valors que ensenyeu.

2. Per a cada patró llistat*, calcular:

   2.1 El nombre de classes que contenen com a mínim un ORF amb el patró indicat a la seva descripció.

2.2 El nombre mitjà d'ORFs amb els quals es relacionen els ORFs amb el patró indicat a la seva descripció.

* Els patrons pels quals caldrà resoldre els càlculs 2.1 i 2.2 són:

  - La descripció conté el terme *protein*. Per exemple, l'ORF amb descripció *electron transfer flavoprotein alpha subunit* encaixaria amb aquesta definició.
  - La descripció conté una paraula de 13 caràcters i aquesta conté el terme *hydro*. Per exemple, l'ORF amb descripció *3-hydroxyacyl-CoA dehydrogenase* encaixaria amb aquesta definició.

3. Per a cada enter M entre 2 i 9 (ambdós inclosos), calcula el nombre de classes que tenen com a mínim una *dimensió* major estricte (>) que 0 i alhora múltiple de M. Amb el terme dimensió ens referim a cadascun dels 4 números que formen l'identificador de la classe (explicat a la secció anterior). Aquest càlcul haurà de resultar en quelcom com:

   ```
   M=2: ? classes
   M=3: ? classes
   ...
   M=9: ? classes
   ```

   on ? repesenta un enter.

A més, haureu de generar codi que permeti representar tots els resultats gràficament (excepte pel càlcul 1.1). Per a cada funció caldrà que penseu i justifiqueu quin tipus de gràfica és la més adient per a representar el resultat.

El codi haurà d'estar correctament comentat, incloent-hi documentació de funcions, i correctament testejat usant la llibreria `unittest`. Els testos proporcionats hauran de donar cobertura com a mínim al 50% de la funcionalitat proposada.

3.1 Cobertura dels testos

El mesurament de la cobertura dels testos s'utilitza per avaluar l'eficàcia dels testos proposats. En particular, serveix per determinar la qualitat dels testos desenvolupats i per determinar les parts crítiques del codi que no han estat testejades. Per tal de mesurar aquest valor, proposem l'ús de l'eina `Coverage.py`. A la documentació, podreu trobar com instal · lar-la i com usar-la.

Per a avaluar la qualitat dels testos desenvolupats per la PAC4, demanem un mínim del 50% de cobertura.

4 Ús de Git

Per tal de posar en pràctica el que heu après a la Unitat 6 sobre `Git`, proposem l'ús de `GitHub Classroom` per a desenvolupar el vostre paquet de Python. GitHub Classroom és una eina gratuïta de codi obert que ajuda a simplificar l'ús educatiu de GitHub. Hem usat GitHub Classroom per a crear una aula com aquesta i on hem creat una tasca per a la PAC4. Per a fer ús d'aquest espai que hem creat, us aconsellem seguir els passos indicats en aquesta guia on s'explica com crear un repositori per a treballar en la tasca que hem preparat. L'enllaç a la tasca el trobareu al missatge de la PAC4 publicat al tauler de l'aula.

L'ús d'aquesta eina no és obligatori per l'avaluació de la PAC4, però creiem que és una molt bona oportunitat per a posar en pràctica els vostres coneixements en un entorn vital per a tothom que treballi o vulgui treballar en l'àmbit de la ciència de dades.

5 Criteris de correcció

Aquesta PAC es valorarà seguint els criteris següents:

- **Funcionalitat** (5 punts): Es valorarà que el codi implementi correctament el que demana l'enunciat.
  - Lectura fitxers (1 punt)
  - Exercici 1 (0.5 punts)
  - Exercici 2 (1.5 punts)
  - Exercici 3 (1 punt)
  - Visualitzacions (1 punt)
- **Documentació** (0.5 punts): Totes les funcions dels exercicis d'aquesta PAC hauran d'estar correctament documentades utilitzant docstrings (en el format que preferiu).
- **Modularitat** (1 punt): Es valorarà la modularitat del codi (tant l'organització del codi en fitxers com la creació de funcions).
- **Estil** (0.5 punts): El codi ha de seguir la guia d'estil de Python (PEP8), exceptuant els casos on fer-ho compliqui la llegibilitat del codi.
- **Tests** (2 punts): El codi ha de contenir una o diverses *suites* de testos que permetin comprovar el bon funcionament de les funcions implementades, obtenint un mínim del 50% de cobertura.
- **Requeriments** (0.5 punts): Hi haurà d'haver un fitxer de requeriments que llisti (només) les llibreries necessàries per a executar el codi.
- **README** i **llicència** (0.5 punts): Es valorarà la creació d'un fitxer de README, que presenti el projecte i expliqui com executar-lo, així com la inclusió de la llicència sota la qual es distribueix el codi (podeu triar la que vulgueu).

5.1 Important

Nota 1: De la mateixa manera que en les PACs anteriors, els criteris transversals es valoraran de manera proporcional a la part de la funcionalitat implementada.

Per exemple, si el codi només implementa la meitat de la funcionalitat demanada, i la documentació d'aquesta part és perfecta, aleshores la puntuació corresponent a la part de documentació seria de 0.25.

Nota 2: És imprescindible que el paquet que lliureu s'executi correctament a la màquina virtual, i que el fitxer de README que inclogueu expliqui clarament com s'ha d'executar el vostre codi per tal de generar les gràfiques resultants de l'anàlisi.

Nota 3: Lliureu el paquet com a un únic arxiu .zip al Registre d'Avaluació Contínua. **El codi de Python haurà d'estar escrit en fitxers plans de Python.**