El repte del SARS-CoV-2

Ara per ara, un dels grans objectius dels científics a nivell mundial és aconseguir una vacuna per al virus SARS-CoV-2. Un dels problemes dels coronavirus és la ràpida mutació del seu ARN. Des de la vessant de la computació, el que es vol aconseguir és classificar les diferents mostres per identificar les diferents classes i crear un arbre genealògic, i a partir d'aquest identificar quines parts del ARN han mutat en cada classe.

L'objectiu del repte (challenge) és classificar les diferents mostres de SARS-CoV-2 que hi ha, i mostrar el seu arbre genealògic.

Requeriments

- Heu de triar els algorismes més apropiats i implementar-los utilitzant les lliberies que hem utilitzat a classe. Per altres llibreries, consulteu al professor.
- El format d'entrada per a la llista de mostres serà CSV, podeu utilitzar la llibreria `csv` que us proporciona python. El format d'entrada de les mostres serà `FASTA`. Tots dos formats són els que utilitza el National Center for Biotechnology Information (NCBI). Podeu consultar-hi tota la informació sobre les seqüències de RNA del SARS-CoV-2.
- El programa principal ha de ser en `python`, al que se li passa per paràmetre el nom del directori que conté les diferents mostres. Els programes secundaris poden ser implementats en `Haskell` o `Rust`.
- Heu de fer un informe que contingui, sobre els algorismes més importants: pseudo-codi, cost teòric i experimental.
- Haureu de crear un projecte `git` en algun repositori públic (github), en el que hi treballareu tot l'equip, i del que em fareu col·laborador (el meu nom d'usuari és `jordiplanes`). Allí hi posareu tot el codi font que aneu fent i també la documentació feta en markdown.

Mòduls de la pràctica

La pràctica tindrà 3 parts diferenciades:

Preprocessament S'hauran d'agafar un subconjunt de totes les mostres que hi ha al repositori NCBI. Haureu d'agafar una mostra de llargada *mediana* de cada país.

Alineament de seqüències S'hauran de comparar les seqüències de dos en dos i establir una mesura de similitut. Com més baix aquest nombre, vol dir que són més semblants, com més alt, menys semblants.

Classificació S'hauran de classificar les seqüències en funció de la seva semblança. Les que són més semblants aniran al mateix conjunt.

Arguments i parametres

L'execució de l'aplicació haurà de seguir la següent sintaxi:

```
$ ./sarscovhierarchy.py <directory>
```

Els arguments del programa són els següents:

<**directory**> el nom del directory que conté els fitxers en format `FASTA`.

Planificació

Us proposo la següent planificació:

Setmana 1 Decidir membres equip, distribuir tasques (cada mòdul o tasca l'haurieu de fer per parelles), crear projecte a `github`, apendre comandes bàsiques de `git`.

Setmana 2 Preprocessament. Anàlisi de les dades a tractar, dissenyar algorisme, estudiar-lo teòricament i implementar-lo, utilitzant la llibreria `csv`. Documentar-ho tot amb markdown.

Setmana 3,4 Alineament de seqüències. Anàlisi dels algorismes existents, anàlisi teòrica de l'algorisme, implementar-lo, i anàlisi experimental. Documentar-ho tot amb markdown.

Setmana 5 Classificació. Anàlisi d'algorismes existents, anàlisi teòrica de l'algorisme, implementar-lo, i anàlisi experimental. Documentar-ho tot amb markdown.

Avaluació

En l'informe de la pràctica, els algorismes principals, les taules i les gràfiques han d'estar comentats.

Els programes han de compilar sense errors ni warnings.

La baremació de la pràctica (sobre 10) serà la següent:

Entorn 1 punt per creació de l'entorn.

Costos 2 punts, anàlisi de costos teòrics i empírics dels algorismes;

Disseny 2 punts, explicació del disseny triat.

Implementació 5 punts. Llenguatges utilitzats, bones pràctiques de programació, utilització dels recursos dels llenguatges.

Enviament

L'assignació és per grups de 3 o 4, i representa un 45% de la nota final. Cadascun dels mòduls d'avaluarà per separat: 15% la primera part, 20% la segona part i 10% la tercera. Presenteu la pràctica al Campus Virtual de la UdL amb dos fitxers: un pdf per a l'informe, i un arxiu comprimit, `tgz` o `zip` (no s'admeten altres formats de compressió), per al codi font.

Els programes que s'hagin de compilar ho han de fer amb:

```
$ make
```

Els programes es testejen fent:

```
$ make test
```

No s'avaluarà cap pràctica que no tingui tots els fitxers per a poder fer la compilació amb `make` i les proves de test amb `make test`.

La pràctica podria ser verificada individualment el primer dia de laboratori després de l'entrega.