Minería de datos bioinformáticos

Objetivos

El objetivo de esta actividad se centra en la aplicación de las herramientas de minería de datos bioinformáticos aprendidas hasta ahora. Entre estas habilidades se encuentran: el manejo de API para realizar consultas a bases de datos biológicas y químicas, así como el uso de bibliotecas especializadas en la manipulación de estos datos.

Pautas de elaboración

Para esta actividad necesitarás un ordenador con Python y Conda instalados.

Accede al siguiente enlace para instalar Python:

<https://www.python.org/downloads/>

Accede al siguiente enlace para instalar Conda:

<https://docs.conda.io/projects/conda/en/latest/user-guide/install/windows.html>

**Actividades**

1. **Obtención de información sobre proteínas y compuestos químicos (6,5 puntos)**

Las API son una herramienta poderosa a la hora de extraer información de las bases de datos públicas. A continuación, se proponen algunas tareas para ponerlas en práctica:

* + A. Crea una función que acepte como entrada una lista de ID del Protein Data Bank y descargue de esta base de datos los archivos formato .cif de cada uno de estos ID. Incluye una forma de controlar cuando un archivo ha podido ser descargado correctamente y cuando no ha sido posible sin que ello afecte a la ejecución del programa.
  + La lista de ID es la siguiente: ['1tup', '2xyz', '3def', '4ogq', '5jkl', '6mno', '7pqr', '8stu', '9vwx', '10yza'] (2,5) **(Script 7 CL5)**
  + B. Toma el ID 1tup y determina su identificador en UniProt a través de su API. Una vez obtengas el identificador, consulta la información en [UniProt](https://www.uniprot.org/), investiga el objeto creado por la llamada y extrae la siguiente información:
    - la fecha de publicación de la entrada,
    - la fecha de la última modificación,
    - si está manualmente revisada (Swiss-Prot) o si no ha sido revisada (Trembl),
    - el nombre del gen y sus sinónimos,
    - el organismo al cual pertenece,
    - el nombre completo de la proteína,
    - la secuencia de aminoácidos de una sola letra
    - Los pdb ids asociados a dicha entrada de UniProt.

Guarda esta información en un DataFrame de Pandas que contenga las siguientes columnas:

* + ['Uniprot\_id', 'Fecha\_publicacion', 'Fecha\_modificacion', 'Revisado', 'Nombre\_del\_gen', 'Sinónimos', 'Organismo', 'PDB\_ids'] (2,5) **(Script 1 - clase 8)**
  + C. Investiga en esta entrada de UniProt de la 1tup si hay información sobre algún cofactor. Indica dónde se identifica al cofactor (Script 4 clases 6 y 7) y utiliza la API de PubChem para extraer información sobre: su identificador de compuesto en Pubchem (cid), su peso molecular exacto, su inchi, inchikey y sobre sus nombres según la iupac. Guarda esta información en un DataFrame que contenga las siguientes columnas:
  + ['Compuesto', 'Pubchem\_id', 'Peso\_molecular', 'Inchi', 'Inchikey', 'Iupac\_name'] (1,5).

1. **Manipulación de datos biológicos (3,5 puntos)**

Biopython y rdkit son librerías muy útiles para manipular datos de origen biológico y químico, respectivamente. Utilízalas para completar estas tareas sobre el archivo 4ogq.cif:

* + A. Parséalo con MMCIFParser() y guarda en una lista todas las heteromoléculas (sin incluir las aguas). (0,5) (Script 4 - clases 6 y 7)
  + B. Parseálo con MMCIF2Dict() en un diccionario, extrae la información sobre la clave '\_pdbx\_IUPAC \_nonpoly' y guarda en un DataFrame la información sobre el nombre de cada una de las heteromoléculas, así como su identificador de tres letras. (0,5) (Script 4 - clases 6 y 7)
  + C. Toma el nombre de cada heteromolécula y mediante la API de Pubchem consigue su SMILES, para los casos que sea posible. Transforma estos SMILES con rdkit en SDF y añade ambas columnas al DataFrame del Apartado 2B. (1)
  + D. Genera un archivo .sdf en el cual guardes todas las moléculas SDF. Añade para cada una de ellas un campo que incluya información sobre su 'Molecular\_weight'. Ayúdate de rdkit para calcular dicho campo y asegúrate que el archivo que has creado puede utilizarse para crear un objeto mol de rdkit con cada una de las moléculas. (1,5)

Extensión y formato

* Los resultados deberán ser entregados como un *script* de Python (formato .py).
* Formato .doc con resultados teóricos y gráficos.
* Incluye los comentarios pertinentes para explicar el código.
* No hay una extensión máxima del *script.*

Rúbrica

|  |  |  |  |
| --- | --- | --- | --- |
| Minería de datos bioinformáticos | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Obtención de información sobre proteínas y compuestos químicos | Obtención de información sobre proteínas y compuestos químicos. | 6,5 | 65 % |
| Manipulación de datos biológicos | Manipulación de datos biológicos | 3,5 | 35 % |
|  |  | **10** | **100 %** |