Challenge 8

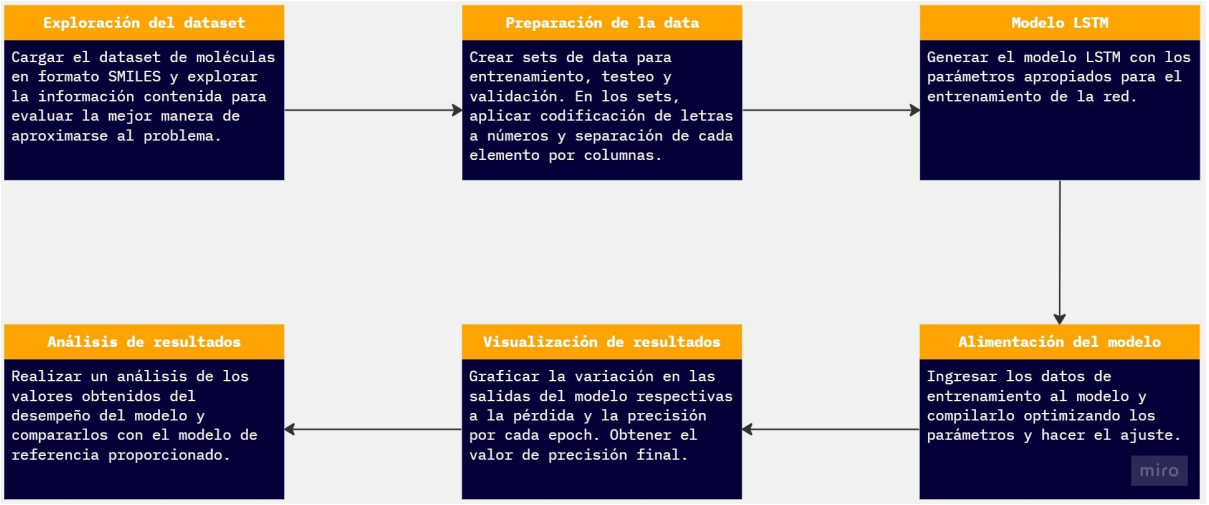
1. **Explicar la arquitectura y funcionamiento de LSTM. Justificar su uso en este trabajo. Opcional: dar sugerencias sobre otras metodologías que pueden ser abordadas.**

   Las LSTM (Long Short-Term Memory) son un tipo especializado de red neuronal recurrente (RNN) diseñadas para modelar y procesar secuencias de datos. Su arquitectura se compone de unidades de memoria llamadas "células LSTM" que están interconectadas para formar una red. Cada célula LSTM tiene tres puertas principales: la puerta de entrada (input gate), la puerta de olvido (forget gate) y la puerta de salida (output gate). Estas puertas controlan el flujo de información dentro de la célula y permiten que las LSTM aprendan a recordar o olvidar información relevante en secuencias largas. Cuando se procesa una secuencia de entrada en una LSTM, la puerta de entrada determina qué nueva información debe agregarse a la memoria de la célula. La puerta de olvido controla qué información anterior debe descartarse de la memoria. Luego, se combina la nueva información con la memoria existente utilizando una operación de suma ponderada. Finalmente, la puerta de salida determina qué información se enviará como salida de la célula LSTM [1].

   El uso de LSTM en el presente trabajo se justifica por la capacidad que posee para modelar secuencias de datos y capturar dependencias a largo plazo. En el contexto de la biología y bioinformática, las moléculas (en este caso, las proteínas) se pueden representar como secuencias de letras (o strings), en las cuales cada letra representa una unidad (en este caso, un aminoácido), lo que las hace adecuadas para ser procesadas por LSTM [1]. Esto simplifica la metodología a seguir, dado que se puede entrenar una red LSTM utilizando un conjunto de datos con información sobre estructuras moleculares y su correspondiente afinidad por la proteasa principal del SARS-COV-2. La LSTM aprenderá patrones y relaciones entre las características moleculares y la afinidad, y podrá generar nuevas moléculas con las propiedades deseadas.

   Además de LSTM, existen otras metodologías que pueden abordar el mismo propósito de encontrar moléculas con alta afinidad por la proteasa principal del SARS-CoV-2. Una de estas metodologías es el uso de redes neuronales convolucionales (CNN). Estas redes aprovechan su capacidad para analizar estructuras moleculares representadas como cuadrículas tridimensionales para aprender a identificar patrones locales y características relevantes en estas estructuras moleculares y las utilizan para predecir la afinidad. Mediante el entrenamiento de la red con conjuntos de datos etiquetados, las CNN pueden generar nuevas moléculas optimizadas con la afinidad deseada [2].

2. **Mediante un diagrama de flujo, explicar los pasos seguidos en este trabajo.**



3. **Dataset: Realizar un análisis exploratorio de los datos y preprocesamiento si lo requiere.**

   Los datos moleculares que emplearemos para entrenar la red neuronal se presentarán en formato SMILES (Simplified Molecular Input Line Entry System). Estos datos fueron limpiados de sales, cargas e información estereoquímica utilizando RDKIT. Todo el código y los datos relacionados están disponibles en el repositorio de GitHub del autor https://github.com/BLarzalere en el archivo **smiles_cleaned.smi.**

   No se requiere preprocesamiento, a continuación se muestra un ejemplo de algunas moléculas en formato SMILES:

```
O=C1Nc2cc(NC(=O)c3c[nH]cc(-c4ccc(C(F)(F)F)cc4)c3=O)ccc2C1=Cc1ccc[nH]1
CC(NC(=O)Nc1cc2[nH]nc(N3CC(C(C)(C)O)C3)c2cn1)c1ccccc1
N=C(N)C1CCCC(NC(=O)CN2CCCCC(NS(=O)(=O)c3ccccc3)C2=O)C1O
CCN1C(C(=O)NC(Cc2ccccc2)C(=O)C(=O)NCCCN2CCCC2=O)Cc2cc3c(cc2S1(=O)=O)OCCO3
COCC(=O)NC1CCC(CCN2CCC(c3cccc4occc34)CC2)CC1
```

4. **Tomando como referencia el repositorio adjunto y el material de apoyo proceder a la generación de moléculas.**

   Para la generación de moléculas primero se ejecutó el entrenamiento del modelo de red neuronal que se encarga de aprender a generarlas. Este fue proporcionado por el profesor del curso en el siguiente archivo **Challenge_8_supp.ipynb.** En este se entrena el modelo por un número limitado de épocas (20 épocas), obteniendo como resultado el modelo entrenado en un archivo de nombre *LSTM_model.h5* (Archivo adjunto).

   Posteriormente se ejecutó el archivo **LSTM Chem - Generate v2.ipynb** de la web de github del autor. El cual nos permite generar las moléculas a partir del modelo entrenado. Antes de realizar el entrenamiento se modificó el código para generar solamente 10 moléculas, debido a que la ejecución completa del archivo llenaba la memoria RAM disponible por el google collab. La modificación realizada se muestra a continuación:

```
# create a for loop to generate molecules based off our sampling dataset's latent
space
```

```
gen_mols, gen_smiles = [], []
for i in range(10): # MODIFICADO
#for i in range(latent_space.shape[0] - 1):
  latent_seed = latent_space[i:i+1]
  sampling_temp = rn.uniform(0.75, 1.26)
  scale = 0.75
  quantity = 20 # MODIFICADO
  # quantity = 50
  mols, smiles = generate(latent_seed, sampling_temp, scale, quantity)
  gen_mols.extend(mols)
  gen_smiles.extend(smiles)
  moles, smiles = [], []
print('SMILES generation completed!')
```

De esta manera se lograron generar un archivo (*generated_smiles_nextgen_v1.csv*, adjunto) con 10 moléculas en formato SMILES nunca antes vistas, las cuales se muestran a continuación:

```
lig_opt_1 NC(C1OC(O)c2noc(C3CCCc4ccc(C(F)(F)F)cc43)c3OCCCC12)C(C)(CC3)N(N)O
lig_opt_2 NC1CC2(O)C3CCCC2C3c2ccc([nH]2)CN2C3=C(Cc4ccccc4Cl)C3C(F)=C1SC2
lig_opt_3 CC12OCCNCCN3CCC(C)(C)C(=O)N1CC(C(F)(F)F)(Oc1ccccc1COc1cc(O)ccc1O2)CC3
lig_opt_4 CCC1CC2N(CC)CC1CNC(=O)CC(O)C[N+](=S)SS=C(Br)c1cccc(OC=O)c1c1cccc(F)c1O2
lig_opt_5 O=C1CC2CCC=CC(=S)N(c3cc4ccccc4[nH]3)C(=O)N2C2CNC(=NC2c2ccc(C(NF)CO)nc2)S1
lig_opt_6 O=C1CCC2(CC=C(C(=CCc3ccccc3F)c3n[s+]([O-])nc3I)c3ccsc3NC=C2S)C(CO)C1(F)F
lig_opt_7 O=C1C2CC(CC1([SH]C=Cc3ccccc3Cl)C(=S)N1N(Cc1cccnc1)CC1CC2F)C(C)(CO)NO
lig_opt_8 CC1(C)C2NCC1Oc1ccoc1C2OP(=O)(c1cnccc1NCc1ccc(F)cc1F)c1cc(C(F)(F)F)c[nH]1
lig_opt_9 CC(C)n1c(C2=CCc3c4[nH]c(=O)c(-c5ccccc5F)nc4=NC3=O)ccc2NC(=O)Cc2ccc([nH]2)c(C)c1
lig_opt_10 CCNCCC2CC1=C2CC(C)(OCCC(F)(C(=O)NC)OCC2CCCC2C)C(NOCc2cncnc2OC#Cc2c(C)noc2C(N)=O)C(OC)c2cccnc2OC1=O
```

5. **Realizar el acoplamiento molecular, guiándose de la referencia. Comparar la energía de afinidad entre las moléculas reportadas en la referencia y las obtenidas en su trabajo.**

   La siguiente tabla resume el resultado del Docking realizado para las moléculas generadas en el apartado anterior, en donde el número más negativo de 'affinity' indica el enlace más fuerte entre el ligando y el receptor (The COVID-19 main protease). Por lo tanto, el ligando 9 es el que mejor se enlaza al receptor, por lo que es la mejor molécula generada por nuestro modelo, considerando nuestras limitaciones.

Lig. Opt.	mode	affinity (kcal/mol)	dist from best mode	
			rmsd l.b.	rmsd u.b.
1	1	-7.9	0.000	0.000
2	1	-8.9	0.000	0.000
3	1	-7.7	0.000	0.000
4	1	-7.5	0.000	0.000
5	1	-8.7	0.000	0.000
6	1	-7.3	0.000	0.000
7	1	-8.6	0.000	0.000
8	1	-8.6	0.000	0.000
9	1	-9.2	0.000	0.000
20	1	-7.2	0.000	0.000

El autor utiliza transfer learning para entrenar un nuevo modelo con sus resultados preliminares, los cuales se muestran a continuación:

Molecule	Binding Affinity
AM-724	-11.70
MO-8	-11.00
AM-728	-11.00
AM-100	-10.60
AM-1517	-10.60
SA-113	-10.50
SA-33	-10.50
AM-1443	-10.40
SA-551	-10.40
AM-726	-10.30

En comparación con sus resultados, los nuestros no se encuentran tan alejados; sin embargo, el transfer learning realizado por el autor mejora dichos resultados, llegando a alcanzar afinidades de hasta -14.6 kcal/mol. Si hubiéramos seguido el mismo procedimiento, es posible que nuestros resultados hubiesen mejorado ligeramente, aunque quizá no tanto como los obtenidos por el autor original, debido a las limitaciones que enfrentamos.

A continuación, mostramos una imagen de cómo se ve la mejor molécula generada por nuestro modelo.

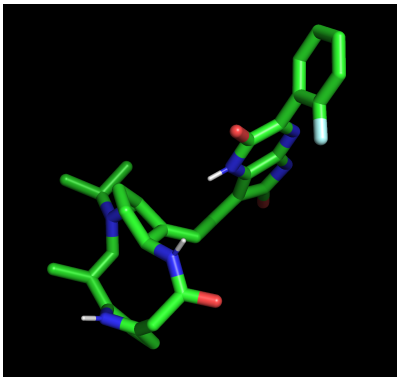


Fig. 1. Lig. Opt. 9

6. **Adjuntar un júpiter notebook y guardar el modelo entrenado.**

*Archivos adjuntos

Referencias

[1] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735. [Online]. Available: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. [Accessed: Jun. 25, 2023]

[2] “De novo design of novel protease inhibitor candidates in the treatment of SARS-CoV-2 using deep learning, docking, and molecular dynamic simulations,” *Comput. Biol. Med.*, vol. 139, p. 104967, Dec. 2021, doi: 10.1016/j.compbiomed.2021.104967. [Online]. Available: http://dx.doi.org/10.1016/j.compbiomed.2021.104967. [Accessed: Jun. 25, 2023]