TECNOLÓGICO DE MONTERREY®

MAESTRÍA EN INTELIGENCIA ARTIFICIAL APLICADA

Materia:
Proyecto Integrador

Avance 2. Ingeniería de características

EQUIPO 29

Miguel Angel Favela Corella	A00818504
Kevin Balderas Sánchez	A01795149
José Manuel García Ogarrio	A01795147

Sponsor del Proyecto

- **Dr. Juan Arturo Nolazco** – Director del Data Science Hub, Tecnológico de Monterrey.

Colaborador Invitado

- **Dr. Carlos Alberto Brizuela Rodríguez** – Investigador, CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California).

1. Antecedentes

La diabetes afecta al 18.3% de la población adulta en México y presenta una tendencia creciente. El desarrollo de fármacos mediante métodos tradicionales requiere entre 10 y 15 años, lo cual limita la respuesta ante esta problemática de salud pública.

El presente proyecto busca acelerar el diseño de péptidos antimicrobianos (PAMs) orientados al tratamiento de la diabetes, mediante el uso del modelo de David Baker, reconocido internacionalmente por su innovación en el diseño de proteínas asistido por Inteligencia Artificial.

El proyecto se desarrolla bajo el patrocinio del Dr. Juan Arturo Nolazco (Tec de Monterrey), con la colaboración del Dr. Carlos Alberto Brizuela Rodríguez (CICESE), uniendo experticia en data science, modelado computacional y biología estructural.

2. Entendimiento del negocio

2.1 Formulación del problema

¿Cómo acelerar el proceso de diseño de péptidos contra la diabetes utilizando IA, reduciendo tiempos de desarrollo y minimizando efectos secundarios?

2.2 Contexto

México se enfrenta a una creciente prevalencia de diabetes, lo que genera una carga económica y social considerable. Adoptar enfoques disruptivos como el modelo de David Baker puede transformar el paradigma de descubrimiento de fármacos, haciendo el proceso más ágil y eficiente.

2.3 Objetivos

- Acelerar el diseño de inhibidores peptídicos contra blancos moleculares asociados con la diabetes mediante el modelo de David Baker.
- Validar computacionalmente candidatos prometedores utilizando técnicas de predicción estructural.
- Reducir los costos y tiempos de investigación en comparación con los métodos tradicionales.

2.4 Preguntas clave

- ¿Qué proteínas blanco son relevantes para la diabetes y cómo pueden ser atacadas con PAMs diseñados por IA?
- ¿Qué nivel de precisión alcanza el modelo de David Baker en la predicción de acoplamientos inhibidor-blanco?
- ¿Qué métricas computacionales validan la eficacia de los péptidos generados?
- ¿Qué combinación de parámetros genera los modelos más eficaces?

2.5 Involucrados

- Tec de Monterrey (Data Science Hub): dirección académica y desarrollo de modelos de IA.
- CICESE: soporte en validación biológica y modelado computacional.
- Equipo de maestría: diseño del pipeline, documentación y experimentación.
- Sponsor (Dr. Nolazco): supervisión académica y estratégica.

3. Entendimiento de los datos

3.1 Descripción de los datos

- Secuencias de proteínas vinculadas a la diabetes.
- Información de acoplamientos moleculares.
- Bases de datos de péptidos antimicrobianos ya estudiados.

3.2 Técnica de ML

Supervisado (clasificación y predicción estructural) apoyado en el modelo de David Baker:

- RFDiffusion para el diseño de esqueletos proteicos.
- ProteinMPNN y ESM-IF1 para la generación de secuencias peptídicas.
- AlphaFold 3 para validar los acoplamientos entre inhibidores y blancos moleculares.

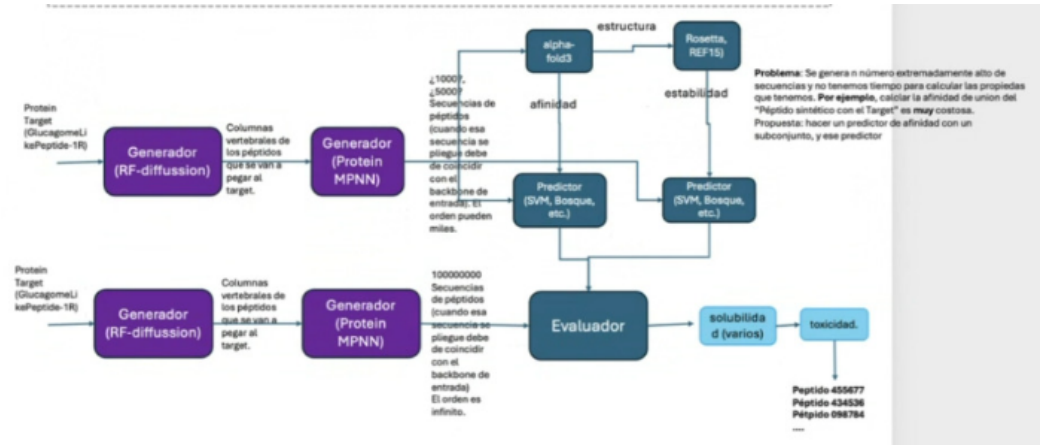
3.3 Identificación de las variables

- Entradas: secuencias de aminoácidos, estructuras 3D de proteínas, datos de acoplamiento.
- Salida: predicción de estabilidad, afinidad y eficacia del péptido candidato frente al blanco molecular.

4. Flujo del proyecto propuesto.

El objetivo del proyecto está basado en varias importantes:

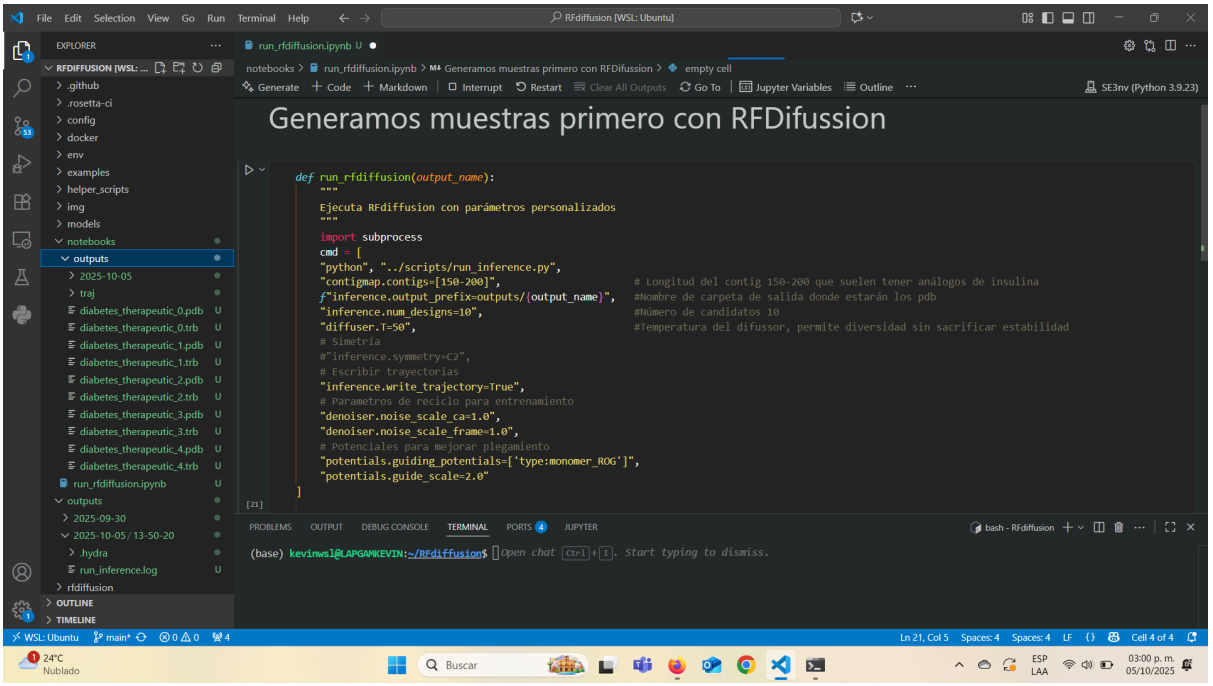
- Generador con RF-difussion (La semana pasada se logró correr de manera local)
- Generador con ProteinMPNN (Esta semana se comprometió a tenerlo de manera local y se logró)
- Validar con AlphaFold (Se trabajará en esto la siguiente semana)
- Validar con Rosetta (Se empezará a trabajar la siguiente semana)
- Entrenar nuevos modelos como SVM, RandomForest, XGBoost, etc para tener un evaluador o predictor robusto y que pueda ser utilizado en una etapa posterior
- Obtener un evaluador final que pueda ser capaz de tomar como entrada de nuevo salidas generadas por RF-difussion y ProteinMPNN y tener como salida estabilidad, solubilidad de diferentes proteínas



4.1. Generador RF-Difussion.

El objetivo de este generador es generar nuevos esqueletos ‘backbone’, es decir, la estructura 3D del esqueleto proteico con solo glicinas, teniendo diversos parámetros de entrada, dichos parámetros estarán en función de características que puedan presentar estas cadenas relacionadas a combatir la diabetes. Como salida se tienen archivos pbd que son típicos en las ciencias biológicas donde.

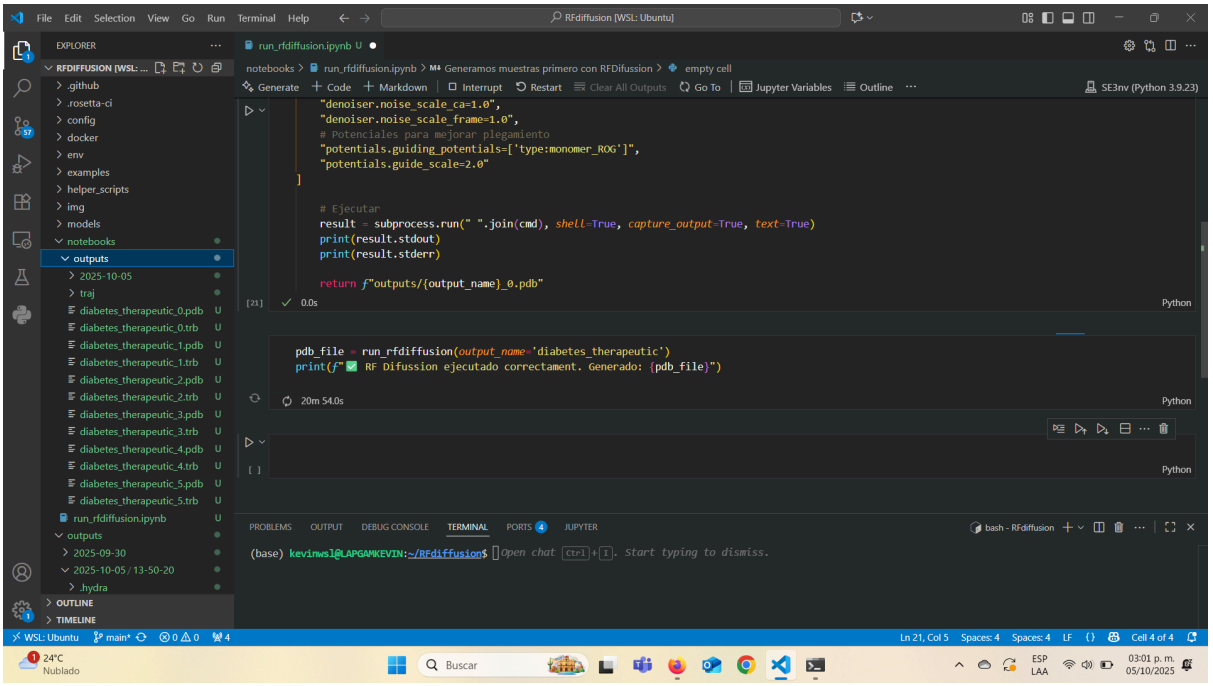
A continuación se presentan algunas capturas del ambiente corriendo de manera local en VSCODE y generando ya los backbones en base a parámetros que ingresamos:



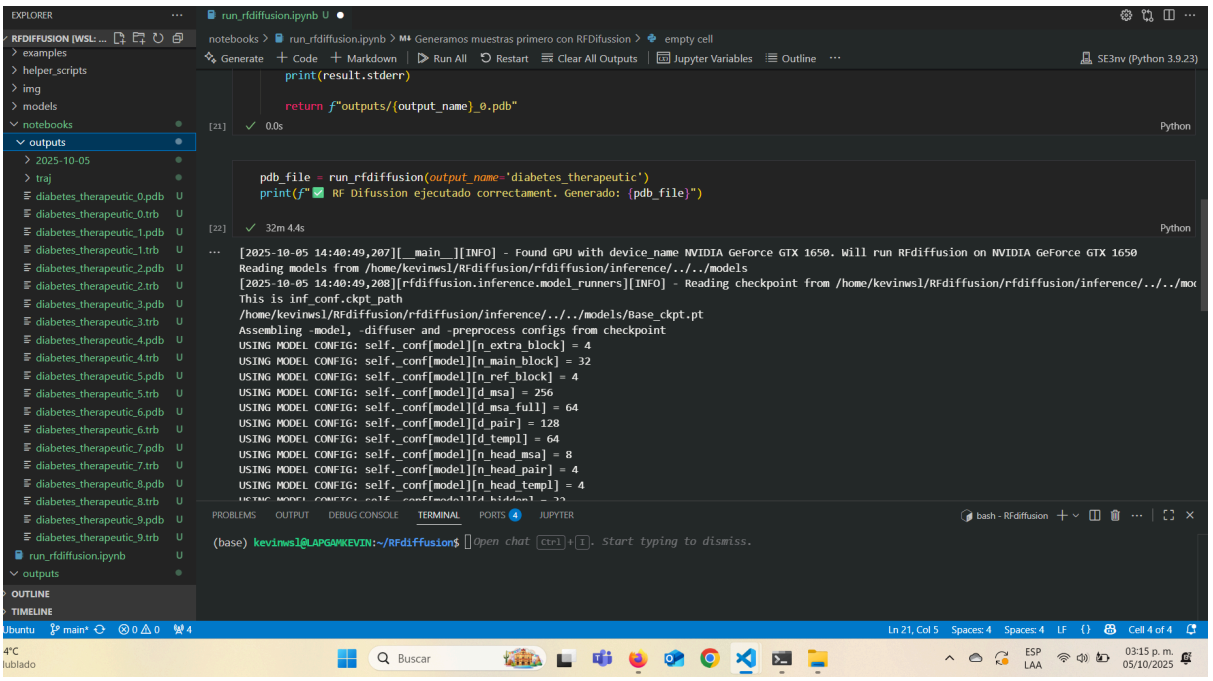
Vemos que toma cerca de 20 minutos tener 5 experimentos de los 10 que pusimos con los siguientes parámetros:

```
"python", "../scripts/run_inference.py",
"contigmap.contigs=[150-200]",                  # Longitud del contig 150-200 que suelen tener análogos de insulina
f"inference.output_prefix=outputs/{output_name}",  #Nombre de carpeta de salida donde estarán los pdb
"inference.num_designs=10",                     #Número de candidatos 10
"diffuser.T=50",                                #Temperatura del difussor, permite diversidad sin sacrificar estabilidad
# Simetría
#"inference.symmetry=C2",
# Escribir trayectorias
"inference.write_trajectory=True",
# Parametros de reciclo para entrenamiento
"denoiser.noise_scale_ca=1.0",
"denoiser.noise_scale_frame=1.0",
# Potenciales para mejorar plegamiento
"potentials.guiding_potentials=['type:monomer_ROG']",
"potentials.guide_scale=2.0"
```

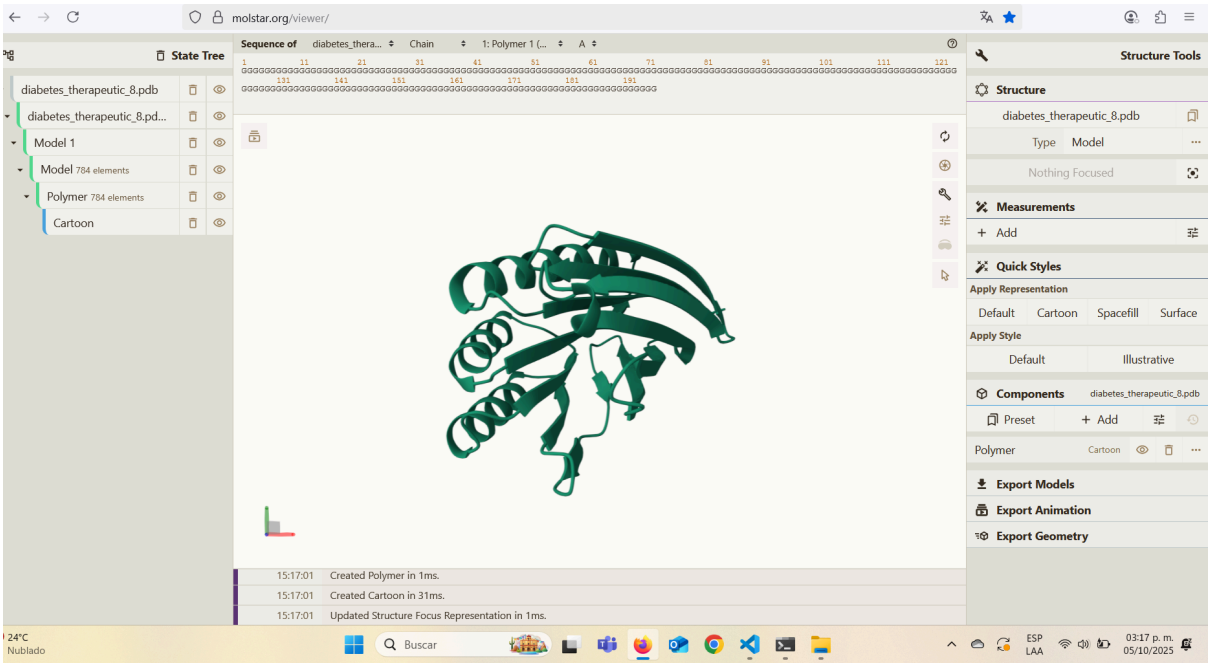
Vemos también que del lado derecho se van generando los archivos .pdb necesarios para futuras operaciones y que se está ejecutando usando el formato de Jupyter Notebook dentro de VSCODE. Asimismo, se debe tener en cuenta todavía en un futuro y con discusiones que tendremos con los asesores del proyecto se trabajará en una selección de parámetros óptima para nuestros fines de combatir la diabetes, resaltando que el objetivo de tener esto de manera local se cumplió en tiempo y forma.



Proceso terminado con las 10 muestras generadas, utilizando una GPU GeForce GTX 1650 tomó cerca de ~35 minutos:



Por último, comprobamos en MolViewer, que es un visualizador de este tipo de archivos pdb para ver cómo se aprecia lo que generó el RF-difussion



4.2. Generador ProteinMPN

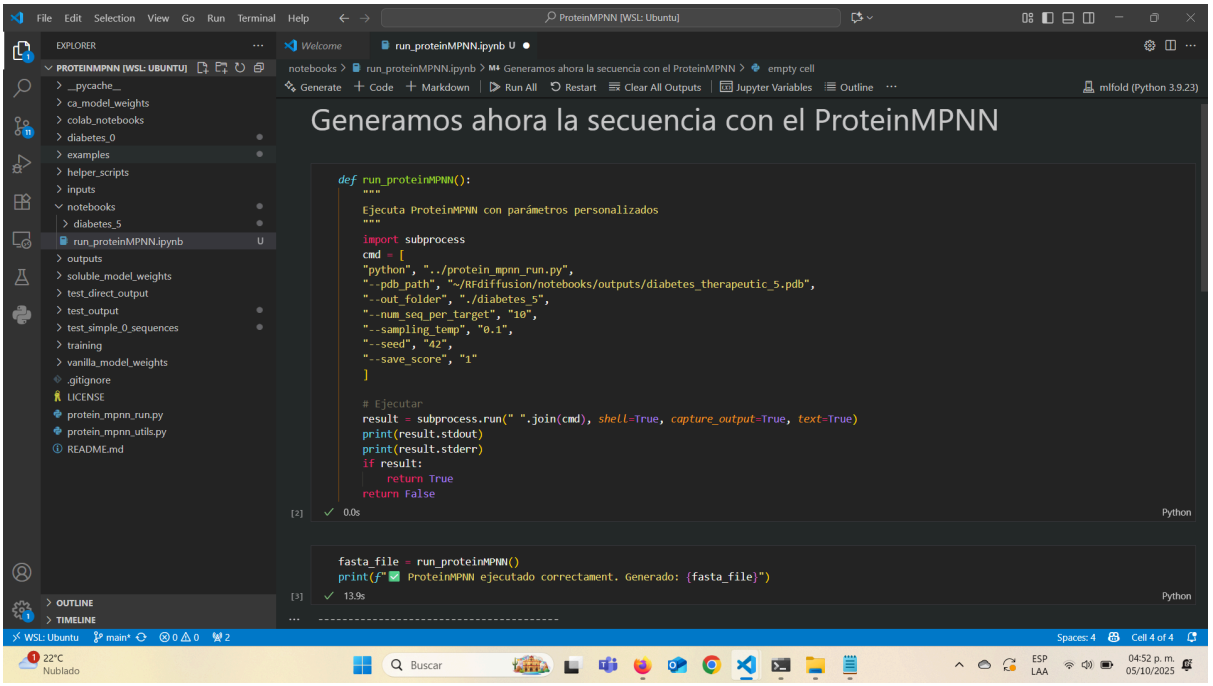
Luego, partiendo de dichos archivos pbd ahora se procede a encontrar la cadena que mejor represente dicha estructura, dicha cadena la genera ahora el ProteinMPN obteniendo así la secuencia completa de aminoácidos que debería adoptar esa estructura. Primero que nada es necesario que tengamos la ruta de nuestro local donde se encuentran los archivos .pdb generados del paso anterior.

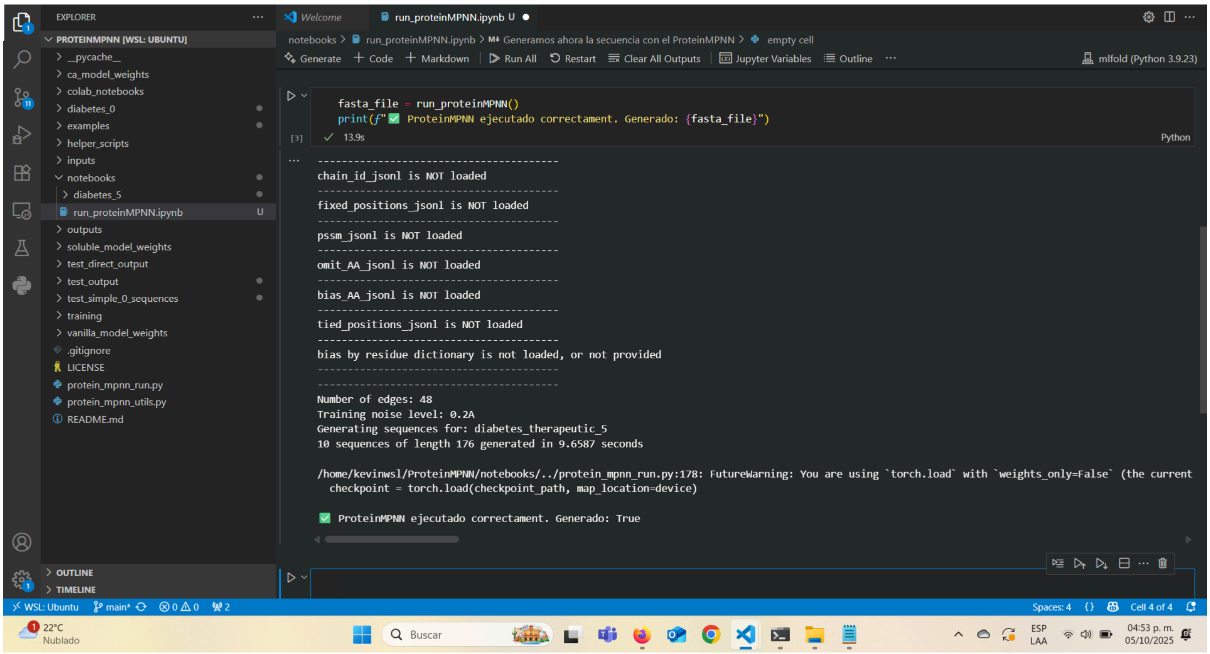
```
(SE3nv) kevinwsl@LAPGAMKEVIN:~/RFdiffusion/notebooks/outputs$ ls
2025-10-05                  diabetes_therapeutic_2.trb  diabetes_therapeutic_5.trb  diabetes_therapeutic_8.trb
diabetes_therapeutic_0.pdb  diabetes_therapeutic_3.pdb  diabetes_therapeutic_6.pdb  diabetes_therapeutic_9.pdb
diabetes_therapeutic_0.trb  diabetes_therapeutic_3.trb  diabetes_therapeutic_6.trb  diabetes_therapeutic_9.trb
diabetes_therapeutic_1.pdb  diabetes_therapeutic_4.pdb  diabetes_therapeutic_7.pdb  run_rfdiffusion_report.pdf
diabetes_therapeutic_1.trb  diabetes_therapeutic_4.trb  diabetes_therapeutic_7.trb  traj
diabetes_therapeutic_2.pdb  diabetes_therapeutic_5.pdb  diabetes_therapeutic_8.pdb
(SE3nv) kevinwsl@LAPGAMKEVIN:~/RFdiffusion/notebooks/outputs$ pwd
/home/kevinwsl/RFdiffusion/notebooks/outputs
```

Luego de eso, es necesario activemos el ambiente de conda encargado de hacer el ProteinMPNN y ejecutar el script “python protein_mpnn_run.py”:

```
(mlfold) kevinwsl@LAPGAMKEVIN:~/ProteinMPNN$ python protein_mpnn_run.py     --pdb_path ~/RFdiffusion/notebooks/outputs/diabete
s_therapeutic_0.pdb     --out_folder ./test_simple_0_sequences     --num_seq_per_target 10     --sampling_temp "0.
1"     --seed 42     --save_score 1
----------------------------------------
chain_id_jsonl is NOT loaded
----------------------------------------
fixed_positions_jsonl is NOT loaded
----------------------------------------
pssm_jsonl is NOT loaded
----------------------------------------
omit_AA_jsonl is NOT loaded
----------------------------------------
bias_AA_jsonl is NOT loaded
----------------------------------------
tied_positions_jsonl is NOT loaded
----------------------------------------
bias by residue dictionary is not loaded, or not provided
----------------------------------------
/home/kevinwsl/ProteinMPNN/protein_mpnn_run.py:178: FutureWarning: You are using `torch.load` with `weights_only=False` (the c
urrent default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data whic
h will execute arbitrary code during unpickling (See https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models
 for more details). In a future release, the default value for `weights_only` will be flipped to `True`. This limits the funct
ions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless t
hey are explicitly allowlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights
_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issu
es related to this experimental feature.
  checkpoint = torch.load(checkpoint_path, map_location=device)
----------------------------------------
Number of edges: 48
Training noise level: 0.2A
Generating sequences for: diabetes_therapeutic_0
10 sequences of length 182 generated in 8.915 seconds
(mlfold) kevinwsl@LAPGAMKEVIN:~/ProteinMPNN$
```

Vemos que se ejecutó de manera correcta el script, ahora vamos a poner eso también en un notebook para verificar una vez más. Observamos el resultado:





Y como luego de correr el ProteinMPNN tenemos los archivos FASTA, que son ya las secuencias generadas para el backbone del paso anterior, lo comprobamos con:

```
(mlfold) kevinwsl@LAPGAMKEVIN:~/ProteinMPNN$ cat ./notebooks/diabetes_5/seqs/*.fa
>diabetes_therapeutic_5, score=2.1227, global_score=2.1227, fixed_chains=[], designed_chains=['A'], model_name=v_48_020, git_hash=8907e6671bfbfc92303b5f79c4b5e6ce47cdef57, seed=42
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
>T=0.1, sample=1, score=1.2255, global_score=1.2255, seq_recovery=0.0057
AALAAELAAELAEEEEELEEEEEEEIEEEMEEIKELVEKVEELIEEKLKDLPEETKEYVKKRLLKLFIKWVKIFEELYLEKKEPFETVVDRCMELVRELLLELRRRGFDEETIEKVIECCKELMELILEYFFEKISPEENKELKETLKKKWEELTP
VLLAEVEAEAAAAAAAAALA
>T=0.1, sample=2, score=1.1009, global_score=1.1009, seq_recovery=0.0057
MEKLKELEEKKKEEEKKEEEKEEKEIEKEKEKYKELIEKVEKMIEEILKDLPEEVKEKVKKRLLKLFIKWLKIFEEEYLKKKKSFEEVIEKIMEEFKKLLIELKKEGFDEETLEKVKEACKEILKILLEYFSEKLSKEEVEKLKKILEEKIEKEWP
KIEKEVEEEIEKEKKEKEEK
>T=0.1, sample=3, score=1.2079, global_score=1.2079, seq_recovery=0.0114
AALAAALAAALAAAAAAAEAARAEEVEALLEEHAELMAEVEALLDELLADLPEETRARLRSRLLRLFVEMLRLWERLYLELGEPFETVVARHMELHRELLLELRRAGASEEELDKAERALRELARLLLEHFAALLSPEERAEARAILERTWAERLP
VLRAEVAAEAAAEEAARAAA
>T=0.1, sample=4, score=1.1906, global_score=1.1906, seq_recovery=0.0114
AAAAAAAAAAAAAAAAAAAAARAAEIAALQAEHAELMAKVEALIAALLAGLPAATKARLSSRLRALFVEWLQIWETEYLENKKPFEEVVAKIMALVRELLRELRAAGADDATLAVAVQACRELMKLLLEHFSKLLSPEEVEELKKTLEETWEKETP
KLLAEVAAEQAAAEAAAAAA
>T=0.1, sample=5, score=1.1162, global_score=1.1162, seq_recovery=0.0057
LEELLEELEEEEEEEEEEEEEEEEIEKELEKIKELTEKVKKKIEEILKDLPEEKKEKIKKRLLKLFKKWLEIWEKLYLEEKKSFEEVVEECLELAKELLLELKEEGFDEETLEKVEEAIRELFELILEYFSEKLSPEEVEKAREILEKKLAEKFP
EIREEVEEEIEEEEEEEEEL
>T=0.1, sample=6, score=1.1905, global_score=1.1905, seq_recovery=0.0057
SAAALAAALAAAAARAAAEAAEEAEDEEEWEKHAELMEEVEALLDELLADLPEETRARLRRRLLRLFLRWLRIFEEEYLKEKKSFEEVVDKIMEEVLALLLELRAEGLDEETLEVVEEACERLMELLLEHFSEKLSEEENEKLRKILKEKWEKEMP
KLRARVAAAAAAAAAAAAAA
>T=0.1, sample=7, score=1.2303, global_score=1.2303, seq_recovery=0.0057
AAAALAAAAALAAALAAAEAAEEAEIEEEMEEIKELMEKVEELLEKLTASLDEETKAEVTKALKRLFVEMLKIFETDYLKNKKPFEEVMLKVMELFRELLLRFAEAGFDEETLAVVVKACRELLELLLEYFSKYLSPEEVEKLKKTMEELWKEKTP
VLLAEVAAERAAREAAAAAA
>T=0.1, sample=8, score=1.1866, global_score=1.1866, seq_recovery=0.0170
AALLLAAAAAAAAAAAAAAARAAAAEAEAAERAALTARVLALLDRLLASLPPETRARLRRRLLALFLEWLEIFEEEYLEEGKPFEEVMEEVMELFRELLLEARREGLDEETLEVLVEACRELGRLLLEELKELLSEEENEELEKIWEETWEEKLP
VLREEVERELREEEAARAAA
>T=0.1, sample=9, score=1.2158, global_score=1.2158, seq_recovery=0.0114
LEELLELLELLEKKKEELEKKKEEEDEKKWEEHKEFIEKLKKLIDEILKSLDEETRKKVKKRLLRLALEWLEIFETEYLKEGKPFEEVIDKIMEKFEELLLELKEEGFSEEVLEKVEEACYKILELIIEYFSKYLSKEEVEKLKKTFKEKWEKKLP
VIREKVEKRQEEREKLLKEL
>T=0.1, sample=10, score=1.2165, global_score=1.2165, seq_recovery=0.0057
LAELEEELEELAAELEELEEALEEEYEKEAEKIKELMEKVEKMLEEILKDLPEEVKEVLTKRLKRYFIEALKIFEKEYLEEKKSFEEVVEKILELQLELLKELRREGFSEEVLEKAEEALKEILEILLEYFSKYLSEEEVKELKEILEKAWKELTP
VLKKEAEEELKREEELLKAL
```

Viendo así las cadenas generadas y los scores que van en un rango de 1.009 - 1.2303, mencionando que los scores no tienen un umbral bueno o malo, derivado de que son valores de una función llamada log-probs negativos.

4.3. Próximo Paso : Validar archivo FASTA con AlphaFold y Rosetta (Siguiente Semana).

5. Bibliografía

- Jumper, J., Evans, R., Pritzel, A., et al. (2021). *Highly accurate protein structure prediction with AlphaFold*. Nature, 596(7873), 583–589.
- Watson, J. L., Juergens, D., Bennett, N. R., et al. (2023). *De novo design of protein structure and function with RFdiffusion*. bioRxiv. https://doi.org/10.1101/2023.03.20.533421
- Anand, N., & Achim, T. (2022). *ProteinMPNN: A deep learning model for protein sequence design*. bioRxiv. https://doi.org/10.1101/2022.07.20.500902